#4
10/18/02

Docket No.: 50006-140 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	
Tomohiro MORIMURA, et al.	:	
Serial No.:	:	Group Art Unit:
Filed: March 01, 2002	:	Examiner:

For: MULTI-PROCESSOR SYSTEM APPARATUS

CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Commissioner for Patents
Washington, DC 20231

Sir:

In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claim the priority of:

Japanese Patent Application No. 2001-056475, filed March 1, 2001

A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY

Stephen A. Becker
Registration No. 26,527

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 SAB:mlw
Date: March 1, 2002
Facsimile: (202) 756-8087

534472 (45)
50006-140
Tomohiro MORIMURA et.

日　本　国　特　許　庁 March 1, 2002

JAPAN PATENT OFFICE

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

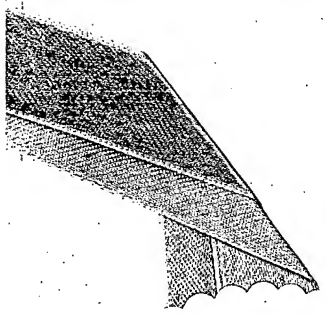
This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年　３月　１日

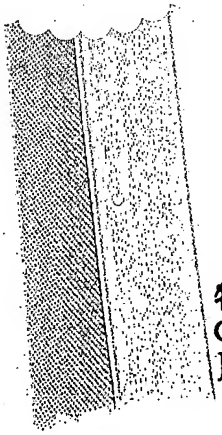
出願番号
Application Number: 特願２００１－０５６４７５

[ST.10/C]: [ＪＰ２００１－０５６４７５]

出願人
Applicant(s): 株式会社半導体理工学研究センター



CERTIFIED COPY OF
PRIORITY DOCUMENT



２００２年　１月２５日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造

【書類名】　特許願
【整理番号】　172781
【提出日】　平成13年 3月 1日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 15/16
【発明者】
　【住所又は居所】　神奈川県横浜市港北区日吉３－６－８　グランブル日吉１０１号室
　【氏名】　森村　知弘
【発明者】
　【住所又は居所】　神奈川県横浜市港北区日吉７－１６－２７－１０３
　【氏名】　天野　英晴
【特許出願人】
　【識別番号】　396023993
　【住所又は居所】　東京都港区新橋６丁目１６番１０号
　【氏名又は名称】　株式会社半導体理工学研究センター
【代理人】
　【識別番号】　100062144
　【弁理士】
　【氏名又は名称】　青山　葆
【選任した代理人】
　【識別番号】　100086405
　【弁理士】
　【氏名又は名称】　河宮　治
【手数料の表示】
　【予納台帳番号】　013262
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9608010

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　マルチプロセッサシステム装置

【特許請求の範囲】

【請求項１】　複数のプロセッサがネットワークを介して相互に接続されてなるマルチプロセッサシステム装置において、

プロセッサ、メモリ部及び上記ネットワークとのインタフェースを行うインタフェース部からなる複数のプロセッサエレメントと、

該各プロセッサエレメント間の接続を行う多段のスイッチによって構成された、階層構造を有する多段結合網と、

を備え、

上記各プロセッサエレメント及び多段結合網は、所定の数を基数とした階層構造にクラスタリングされると共に、各時刻ごとに生成された上記多段結合網における各スイッチの状態を示すスイッチ状態表を用いてあらかじめ静的にスケジューリングされたスケジュールに基づいて、プロセッサエレメント間のパケット転送を行うことを特徴とするマルチプロセッサシステム装置。

【請求項２】　上記階層構造を有する多段結合網は、下位階層から上位階層にパケット転送を行うアップストリーム用の結合網と、上位階層から下位階層にパケット転送を行うダウンストリーム用の結合網とをそれぞれ備えることを特徴とする請求項１記載のマルチプロセッサシステム装置。

【請求項３】　上記スイッチ状態表は、各スイッチごとの、出力端子を保持しているパケットの情報と、該出力端子を要求しているパケットの情報と、該出力端子の状態を示した情報とで構成されることを特徴とする請求項１又は２記載のマルチプロセッサシステム装置。

【請求項４】　上記各プロセッサエレメント及び多段結合網は、１つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、他の時刻のスイッチ状態表で該出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うことを特徴とする請求項１，２又は３記載のマルチプロセッサシステム装置。

【請求項５】　上記多段結合網は、クロス網であり、上記各プロセッサエレメント及び多段結合網は、１つのクロス網内のパケット転送時に、１つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、パケットの要求がない他のスイッチの出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うことを特徴とする請求項４記載のマルチプロセッサシステム装置。

【請求項６】　上記各パケットに対するスケジューリングは、コンパイラによってあらかじめ行われることを特徴とする請求項１、２、３、４又は５記載のマルチプロセッサシステム装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、複数のプロセッサを使用したマルチプロセッサシステム装置に関し、特に、多数のプロセッサとメモリモジュールを多段のスイッチ（多段結合網）によって接続する構成をなすマルチプロセッサシステム装置に関する。

【０００２】

【従来の技術】

多数のプロセッサとメモリモジュールをスイッチによって接続する構成を有するマルチプロセッサシステム装置において、１つのスイッチに複数のパケットが集中して衝突した場合、データ処理に時間を要しデータ処理性能が低下するという問題があった。このため、スイッチにおけるパケットの衝突を低減することができるノンブロッキング網、リアレンジブル網及びブロッキング網が提案されていた。

【０００３】

ノンブロッキング網は、クロスバ網やクロス(Clos)網等があり、スケジューリングによって出線競合を回避すればスイッチ内では衝突が起きることはない。また、リアレンジブル網は、それぞれのスイッチ構成要素の設定をスケジュールすることによって衝突をなくすことができる。一方、ブロッキング網は、一般的に

はスケジューリングによって衝突をなくすことができないが、一定のアクセスパターンに対してはスケジューリングによって衝突をなくすことができる。

【０００４】

【発明が解決しようとする課題】

しかし、ノンブロッキング網では、プロセッサ及びメモリモジュールの数に対するハードウェア量の増加が大きいため、大規模なシステムに使用するとコストが膨大なものとなる。また、リアレンジブル網は、ノンブロッキング網と比較してハードウェアのコストは小さいが、スケジューリングに要する時間が多大なものとなり、マルチプロセッサでの利用は困難であった。更に、従来のブロッキング網のスケジューリングは、一定のアクセスパターンの並べ換えに対してのみ無衝突にする方法であり、マルチプロセッサで実際に利用することができるのは、配列要素が一定の順番に並んでいる特殊な場合に制限されていた。

【０００５】

本発明は、上記のような問題を解決するためになされたものであり、多数のマルチプロセッサとメモリモジュールを使用した大規模なシステムに対して、コンパイラが容易に静的スケジューリングを行うことができ、一般的な同時アクセスパターンに対して無衝突なパケット転送を実現することができるマルチプロセッサシステム装置を得ることを目的とする。

【０００６】

【課題を解決するための手段】

この発明に係るマルチプロセッサシステム装置は、複数のプロセッサが所定のネットワークを介して相互に接続されてなるマルチプロセッサシステム装置において、プロセッサ、メモリ部及びネットワークとのインタフェースを行うインタフェース部からなる複数のプロセッサエレメントと、該各プロセッサエレメント間の接続を行う多段のスイッチによって構成された、階層構造を有する多段結合網とを備え、各プロセッサエレメント及び多段結合網は、所定の数を基数とした階層構造にクラスタリングされると共に、各時刻ごとに生成された多段結合網における各スイッチの状態を示すスイッチ状態表を用いてあらかじめ静的にスケジューリングされたスケジュールに基づいて、プロセッサエレメント間のパケット

転送を行うものである。

【０００７】

また、上記階層構造を有する多段結合網は、下位階層から上位階層にパケット転送を行うアップストリーム用の結合網と、上位階層から下位階層にパケット転送を行うダウンストリーム用の結合網とをそれぞれ備えるようにしてもよい。

【０００８】

具体的には、上記スイッチ状態表は、各スイッチごとの、出力端子を保持しているパケットの情報と、該出力端子を要求しているパケットの情報と、該出力端子の状態を示した情報とで構成されるようにした。

【０００９】

また、上記各プロセッサエレメント及び多段結合網は、１つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、他の時刻のスイッチ状態表で該出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うようにした。

【００１０】

一方、上記多段結合網がクロス網の場合、各プロセッサエレメント及び多段結合網は、１つのクロス網内のパケット転送時に、１つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、パケットの要求がない他のスイッチの出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うようにしてもよい。

【００１１】

具体的には、上記各パケットに対するスケジューリングを、コンパイラによってあらかじめ行うようにした。

【００１２】

【発明の実施の形態】

次に、図面に示す実施の形態に基づいて、本発明を詳細に説明する。

第１の実施の形態.

図１は、本発明の第１の実施の形態におけるマルチプロセッサシステム装置の例を示した概略のブロック図である。

図１において、マルチプロセッサシステム装置１は、数十から数千程度のプロセッサエレメントＰＥを、階層構造をもつ多段結合網(Multistage Interconnection Network:MIN)で接続してなる。なお、図１では、３階層の場合を例にして示している。

【００１３】

マルチプロセッサシステム装置１は、クラスタＤ０～Ｄｘ（ｘは、ｘ＞０の整数）、及び該各クラスタＤ０～Ｄｘ間の接続を行う結合網(Interconnection Network)Ｅ０で構成されている。更に、各クラスタＤ０～Ｄｘは、クラスタＡ０～Ａｎ（ｎは、ｎ＞０の整数）、及び該各クラスタＡ０～Ａｎ間の接続を行う結合網Ｃ０～Ｃｘでそれぞれ構成されている。また、各クラスタＡ０～Ａｎは、プロセッサエレメントＰＥ０～ＰＥｍ（ｍは、ｍ＞０の整数）、及び該各プロセッサエレメントＰＥ０～ＰＥｍ間の接続を行う結合網Ｂ０～Ｂｎでそれぞれ構成されている。

【００１４】

すなわち、マルチプロセッサシステム装置１では、数百～数千のプロセッサエレメントＰＥを結合するためにネットワークを階層構造に分離し、十数～数十のプロセッサエレメントＰＥを、中規模なマルチプロセッサシステム装置に採用される多段結合網という複数のスイッチを数段にわたって接続し、目的のプロセッサエレメントＰＥへは、途中のスイッチの切り換えによって転送経路が形成されている。

【００１５】

プロセッサエレメントＰＥ０～ＰＥｍはそれぞれ同じ構成であることからプロセッサエレメントＰＥｉ（ｉ＝０～ｍ)を例にして説明する。

図２は、プロセッサエレメントＰＥｉの構成例を示した概略のブロック図である。

図２において、プロセッサエレメントＰＥｉは、プロセッサＰＵ、メモリＭＥ及びネットワークインタフェースＮＩで構成されている。プロセッサＰＵとメモ

リＭＥは接続され、更にプロセッサＰＵ及びメモリＭＥはネットワークインタフェースＮＩを介して対応する結合網Ｂｉに接続されている。

【００１６】

このような構成において、各クラスタＡ０～Ａｎにおける同一クラスタ内のプロセッサエレメント間の接続を行う構成をレベル０とし、各クラスタＡ０～Ａｎ間での接続を行う構成をレベル１とし、各クラスタＤ０～Ｄｘ間での接続を行う構成をレベル２とする。すなわち、各クラスタＡ０～Ａｎがレベル０であり、各クラスタＤ０～Ｄｘがレベル１であり、結合網Ｅ０がレベル２となり、レベル０～２の３階層を形成している。言いかえれば、各クラスタＤ０～Ｄｘ及び結合網Ｅ０はクラスタＦ０とすることができ、クラスタＦ０がレベル２となる。

【００１７】

ここで、一般的な多段結合網の１つであるクロス(Clos)網の例を図３に示す。

クロス網では、各段のスイッチの役割から１段目をディストリビュータ(distributor)、２段目をエクスチェンジャ(exchanger)、３段目をコンセントレータ(concentrator)と呼ぶ。なお、図３では、４入力４出力のスイッチを使用し、４つの該スイッチで各段を構成した場合を例にして示している。

【００１８】

多段結合網は、結合しているノードの数、すなわちマルチプロセッサシステム装置ではプロセッサエレメントＰＥの数(ｍ＋１)と構成要素のスイッチにおける入力端子又は出力端子の数ｋによって、すべてのプロセッサエレメントＰＥに対して転送経路が形成可能となる段数は、$\log_k(m+1)$となる。図３では、相対するプロセッサエレメントＰＥは、同じプロセッサエレメントを示していることからｍ＋１＝１６となりｋ＝４である。

【００１９】

このため、スイッチを２段介することによってすべてのプロセッサエレメントＰＥへの転送経路が形成されるが、より大きな転送容量を得ると共に転送経路に冗長性を持たせるために３段のスイッチによってクロス網が構成されている。すなわち、１つのプロセッサエレメントＰＥには、ディストリビュータをなすスイッチエレメントの１つの入力端子とコンセントレータをなすスイッチエレメント

の１つの出力端子が対応して接続されている。

【００２０】

このような多段結合網は、構成スイッチの入出力数と段数及び構成スイッチの数によって、ノンブロッキング、リアレンジブル及びブロッキングの３通りに分類することができる。ノンブロッキングは、静的に、転送データの衝突を起こさない転送経路を設定することができ、リアレンジブルは、転送データの衝突が発生した場合に、転送経路を再設定することによって無衝突な経路を形成することができる。ブロッキングは、転送データの衝突が発生した場合に、転送経路を再設定しても無衝突な経路を形成することができない。例えば、図３で示したクロス網では、構成スイッチの入力端子数又は出力端子数をｋとし、中間段のスイッチ数をｐとすると、ｐ＞(２ｋ－１)のときはノンブロッキング、ｐ≧ｋのときはリアレンジブル、ｐ＜ｋのときはブロッキングとなる。

【００２１】

一方、多段結合網で数百～数千ものプロセッサエレメントＰＥを接続することは、ハードウェア的に困難で現実的ではない。このため、数個のプロセッサエレメントＰＥをクロスバ(Crossbar)スイッチに接続してレベル０のネットワークとすると共に、該クロスバスイッチを入力とする多段結合網で十数～数十のプロセッサエレメントＰＥを結合してレベル１のネットワークとする。更に、複数の該多段結合網を結合するための拡張段を複数のスイッチで形成してレベル２のネットワークとする。

【００２２】

同様に、システム装置の規模に応じた階層の拡張段を付加することによって大規模なシステム装置を相互結合させて、多段結合網を基本網とした階層構造に拡張させることにより、スケーラビリティを得ることができる。このように、各階層のネットワークは、１つのサブネットワークとしてとらえることが可能であるため、各階層レベルに応じてレベルｓ（ｓは、ｓ＞０の整数)のネットワークＮＷｓと呼ぶ。

【００２３】

このような実例として、クロス網を基本多段結合網とした階層構造ネットワー

クについて説明する。

図４及び図５は、基本網となるクロス網内における階層サブネットワークの例を示した図であり、図４は、クロス網内におけるレベル０のネットワークの例を、図５は、クロス網内におけるレベル１のネットワークの例を示している。なお、図４及び図５では、４つのプロセッサエレメントＰＥ０～ＰＥ３を有する４つのクラスタＡ０～Ａ３を例にして示している。

【００２４】

図４及び図５において、スイッチエレメントＳＤ０～ＳＤ３は、クロス網のディストリビュータをなし、スイッチエレメントＳＥ０～ＳＥ３は、クロス網のエクスチェンジャをなし、スイッチエレメントＳＣ０～ＳＣ３は、クロス網のコンセントレータをなしている。また、スイッチエレメントＳＤ０～ＳＤ３，ＳＥ０～ＳＥ３，ＳＣ０～ＳＣ３は、それぞれ４入力４出力のスイッチエレメントをなしている。

【００２５】

スイッチエレメントＳＤ０及びＳＣ０は、接続された各プロセッサエレメントＰＥ０～ＰＥ３と共にクラスタＡ０を形成し、スイッチエレメントＳＤ１及びＳＣ１は、接続された各プロセッサエレメントＰＥ０～ＰＥ３と共にクラスタＡ１を形成している。同様に、スイッチエレメントＳＤ２及びＳＣ２は、接続された各プロセッサエレメントＰＥ０～ＰＥ３と共にクラスタＡ２を形成し、スイッチエレメントＳＤ３及びＳＣ３は、接続された各プロセッサエレメントＰＥ０～ＰＥ３と共にクラスタＡ３を形成している。

【００２６】

結合網Ｃ０をなすスイッチングエレメントＳＥ０～ＳＥ３が、例えば図４の矢印で示すように入力端子と同じ出力端子へストレートにスイッチングされると、レベル０のネットワークが形成され、クラスタＡ０～Ａ３における同一クラスタ内でのデータ転送が実現する。これに対して、結合網Ｃ０をなすスイッチングエレメントＳＥ０～ＳＥ３が、例えば図５の矢印で示すように入力端子が異なる出力端子にクロスしてスイッチングされると、エクスチェンジャをなす２段目のスイッチエレメントＳＥ０～ＳＥ３が、レベル１のネットワークをなし、クラスタ

Ａ０～Ａ３における異なるクラスタ間でのデータ転送が実現する。

【００２７】

このように、エクスチェンジャをなす２段目のスイッチエレメントＳＥ０～ＳＥ３は、スイッチとしての役割を果たした場合、レベル１のネットワークとして動作し、スイッチとしての役割を果たさなかった場合は、レベル０のネットワークとして動作したことになる。すなわち、１つのクロス網内には、レベル０のネットワークとレベル１のネットワークという２つのサブネットワークが存在することになる。

【００２８】

次に、クロス網間を接続する拡張段、すなわち図１の結合網Ｅ０について説明する。

図６は、階層構造クラスタリングを実現したマルチプロセッサシステム装置１の例を示した図である。なお、図６では、説明を分かりやすくするために、４を基数、すなわち４つのプロセッサエレメントＰＥ０～ＰＥ３を有する４つのクラスタＡ０～Ａ３を備えた４つのクラスタＤ０～Ｄ３で構成される階層構造クラスタリングを実現した場合を例にして示し、プロセッサエレメントは省略して示している。

【００２９】

図６において、１６個のプロセッサエレメントまでをクロス網で直接結合してクラスタＤ０～Ｄ３をそれぞれ形成し、該クロス網同士、すなわちクラスタＤ０～Ｄ３を、付加した拡張段であるレベル２のネットワークにおけるエクスチェンジャをなすスイッチエレメントＳＥａ０～ＳＥａ３を使用して相互結合を行う。該スイッチエレメントＳＥａ０～ＳＥａ３は、それぞれ４入力４出力のスイッチエレメントをなし、図１の結合網Ｅ０をなす。またこの場合、各クロス網内のスイッチエレメントＳＥ０～ＳＥ３は、スイッチエレメントＳＥａ０～ＳＥａ３に接続するための１入力１出力が追加され、５入力５出力となる。

【００３０】

また、更に多くのプロセッサエレメントを結合させる場合は、レベル２のネットワークを相互結合させるためのレベル３のネットワークとしてレベル３のエク

スチェンジャを付加する。すなわち、図１で示したすべての構成を有するクラスタが複数存在し、該各クラスタ間の結合を行う結合網を設けて４階層にする。このように、階層数Ｒである場合、結合されるプロセッサエレメントの数Ｎは、基本網である多段結合網に接続されるプロセッサエレメント数(ｍ＋１)から、下記（１）式のようになる。

$$N=(m+1)\times k^{(R-1)} \quad \cdots\cdots\cdots\cdots\cdots\cdots (1)$$

【００３１】

また、図６のマルチプロセッサシステム装置１において、ｍ＋１＝ｋ×ｋとなることから、上記（１）式は下記（２）式のようになる。

$$N=k\times k\times k^{(R-1)}=k^{(R+1)} \quad \cdots\cdots\cdots\cdots\cdots\cdots (2)$$

【００３２】

次に、上記のような階層構造を有する多段結合網の静的スケジューリング方法について説明する。

階層構造を有する多段結合網において静的スケジューリングする場合の前提条件として、すべてのデータ転送は、コンパイラにおけるスケジューラによって完全に静的解析がなされ、どのタイミングでどこあてのパケットが転送されるという情報が分かっている上でデータアクセスをスケジューリングするものとする。

【００３３】

静的にスケジューリングするためには各時刻におけるスイッチの状態を把握しなければならず、各スイッチエレメントの出力端子ごとに、「現在時刻」、「保持ポート」、「保持クロック」、「ポート要求待ち行列」及び「状態」といった各項目から成り立つスイッチ状態表を作成する。なお、「保持ポート」とは、この出力端子を保持している入力ポート番号であり、「保持クロック」とは、保持しているサイクル(クロック)数である。また、「ポート要求待ち行列」とは、この出力端子を要求している入力端子番号を入れる待ち行列であり、「状態」とは、この出力端子の状態を示しリリース(RELEASED)とホールド(HOLD)の２つの状態がある。

【００３４】

図７は、スイッチ状態表の例を示した図である。なお、図７では、４入力４出

力のスイッチエレメントの場合を例にして示している。

図７において、現在時刻が１５７８４３のときにおける各スイッチの状態を示しており、出力端子＃０が入力端子＃３によって２クロック保持されている。したがって、この２クロックの間は、他の入力端子のパケットが出力端子＃０を獲得することはできない。また、出力端子＃１は開放されているが、入力端子＃０と＃２の各パケットが出力端子＃１の獲得要求を出している。出力端子＃２及び＃３においても開放されているが、入力端子＃１のパケットが出力端子＃２の獲得要求を出している。

【００３５】

各スイッチエレメントにおけるすべてのスイッチにおいて図７で示したようなスイッチ状態表が作成され、コンパイラにおけるスケジューラは該スイッチ状態表に基づいてスケジューリングを行う。図７の出力端子＃１のように、ポート要求待ち行列に２つ以上の獲得要求がある場合は、パケットの優先度に基づいて調停され、調停に負けたパケットのアクセスは後の時刻にずらされる。

【００３６】

一方、調停に勝ったパケットは、出力端子を獲得して保持ポートと保持クロックに記載され、保持クロックが１となる時刻まで保持ポートに記載されると共に状態をホールドにしてスイッチ状態表が作成される。このようなことから、最終的にはすべてのアクセス時間分のスイッチ状態表が必要になるが、ある時刻のアクセスのパケットをスケジュールするときに必要なスイッチ状態表は、該時刻よりも後のものだけであるため、該時刻よりも前の時刻のスイッチ状態表は破棄することができる。

【００３７】

次に、コンパイラによって行われる、スイッチ状態表を使用した静的スケジューリング方法について説明する。なお、ある時刻Ｔsに発行されるパケットの集合Ｕtsを、Ｕts＝ｐ0，ｐ1，……，ｐNと表すこととし、以下静的スケジューリング方法の各処理において、特に明記しない場合はすべてコンパイラによって行われるものである。

【００３８】

まず、パケット集合Ｕtsの要素であるパケットｐｊ（ｊ＝０～Ｎ）に対して、対応するディストリビュータのスイッチエレメントにおけるスイッチ状態表を、パケットのヘッダ(ルーティングタグ等)に応じて作成する。また、スイッチ状態表の現在時刻をＴsに設定する。次に、すべてのパケットｐ1～ｐNに対して、ディストリビュータのスイッチエレメントにおけるスイッチ状態表が作成されると、スイッチ状態表におけるポート要求待ち行列内の入力端子のパケットに対して調停を行う。調停に負けたパケットは、パケット集合Ｕtsから除かれて次の時刻Ｔs+1に発行されるパケット集合Ｕts+1に加えられる。

【００３９】

一方、調停に勝ったパケットに対しては、獲得した出力端子のスイッチ状態表を、保持クロック数分だけ作成又は書き換えを行う。各スイッチエレメントにおけるすべてのスイッチに対する状態が決定すると、出力に応じて対応する次の段のスイッチに対するスイッチ状態表を作成又は書き換えを行う。このときのスイッチ状態表は、現在時刻を１つ進めたときのスイッチ状態を示す。

このような処理を繰り返し、目的地に到着したパケットはその都度パケット集合Ｕtsから取り除かれ、パケット集合Ｕtsが空集合になるまでこのような処理を繰り返す。

【００４０】

上記のような操作によってある時刻Ｔsに発行されたパケットは、無衝突に調整及びスケジューリングされる。また、上記の操作で、パケット集合内に同じノードから発行されるパケットが２つある場合、その内１つは次の時刻のパケット集合に入れられるため、パケットのアクセスが集中している場合は、１つずつパケット集合がずれていくことになる。該スケジューリング対象の時刻Ｔsと同様の処理を全時刻に対して行うことによって静的にパケット転送を完全にスケジューリングすることができる。

【００４１】

図８～図１０は、スイッチ状態表を使用した静的なスケジューリング方法を示したフローチャートであり、図８～図１０を用いて静的なスケジューリング処理の流れについてもう少し詳細に説明する。なお、図８～図１０では、ある時刻Ｔ

sに発行されるパケットの集合Ｕtsを、Ｕts＝ｐ0，ｐ1，……，ｐNと表すこととする。また、図８～図１０の各フローで行われる処理は、特に明記しない限りコンパイラによって行われる。

【００４２】

図８において、まず最初に時刻Ｔsに発行されるパケット集合Ｕtsを１段目の各スイッチの入力端子にエントリする（ステップＳ１）。なお、多段結合網の段数は、入力側から数字を１から昇順に振るものとする。次に、現在注目しているスイッチの段数ＳＴcurを１に設定すると共に、現在処理している最高位の階層数Ｒcurを１に設定する（ステップＳ２）。この後、段数ＳＴcurの各スイッチに対してスケジューリングを行う（ステップＳ３）。段数ＳＴcurの各スイッチに下位階層へのリンクが存在しているか否かを調べ（ステップＳ４）、存在している場合は（ＹＥＳ）、段数ＳＴcurを下位階層の段数に設定し現在時刻Ｔcurを１つ進め（ステップＳ５）、この後ステップＳ３に戻る。

【００４３】

一方、ステップＳ４で、下位階層へのリンクが存在していなかった場合（ＮＯ）、段数ＳＴcurの各スイッチに上位階層へのリンクが存在しているか否かを調べ（ステップＳ６）、存在している場合は（ＹＥＳ）、現在処理している最高位の階層数Ｒcurを１増やすと共に現在注目しているスイッチの段数ＳＴcurを該階層数Ｒcurと同じ数に設定した（ステップＳ７）後、ステップＳ３に戻る。また、ステップＳ６で、上位階層へのリンクが存在していなかった場合（ＮＯ）、本フローは終了する。

【００４４】

ここで、図８のステップＳ３で示したスケジューリング処理について、図９のフローチャートを用いてもう少し詳細に説明する。

図９において、最初に、段数ＳＴcurに属するすべてのスイッチエレメントに対して、入力端子にエントリされたパケットの行き先出力端子番号に基づいて、対応する時刻Ｔcurのスイッチ状態表のポート要求待ち行列に入力端子番号をエントリする（ステップＳ１１）。次に、段数ＳＴcurに属する各スイッチエレメントに対して順に０から番号を振り、注目しているスイッチエレメント番号ＳＷ

curを０に設定する（ステップＳ１２）。

【００４５】

この後、ＳＷcurのスイッチエレメントに対して、スイッチ状態表によるスケジューリングを行い（ステップＳ１３）、現在注目している段数ＳＴcurが最終段であるか否かを調べる（ステップＳ１４）。ステップＳ１４で、最終段である場合（ＹＥＳ）、出力端子にエントリされたパケットを到着パケットとしてパケット集合Ｕtsから削除する（ステップＳ１５）。更に、注目しているスイッチエレメント番号ＳＷcurを１つ進め（ステップＳ１６）、該スイッチエレメント番号ＳＷcurがその段数ＳＴcurの全スイッチ数Ｎst未満であるか否かを調べる（ステップＳ１７）。

【００４６】

ステップＳ１７で、全スイッチ数Ｎst未満である場合（ＹＥＳ）、本フローは終了して図８のステップＳ４に進む。また、ステップＳ１７で、全スイッチ数Ｎst未満でない場合（ＮＯ）、ステップＳ１３に戻る。また、ステップＳ１４で、最終段でなかった場合（ＮＯ）、出力端子にエントリされたパケットを接続されている次の段のスイッチの入力端子にエントリさせ（ステップＳ１８）、ステップＳ１６に進む。

【００４７】

ここで、図９のステップＳ１３で示したスケジューリング処理について、図１０のフローチャートを用いてもう少し詳細に説明する。

図１０において、スイッチ状態表の注目している出力端子番号ＰＯcurを０に設定し（ステップＳ２１）、該出力端子番号ＰＯcurにポート要求待ち行列があるか否かを調べる（ステップＳ２２）。ステップＳ２２で、ポート要求待ち行列がある場合は（ＹＥＳ）、パケットのヘッダ内の優先度に基づいて調停を行い（ステップＳ２３）、ポート要求待ち行列から１つのパケットを選択し（ステップＳ２４）、該パケットが調停に勝ったか否かを調べる（ステップＳ２５）。

【００４８】

ステップＳ２５で、選択したパケットが調停に勝つと（ＹＥＳ）、該パケットを書き込むスイッチ状態表の時刻Ｔhを時刻Ｔcurに設定する（ステップＳ２６）

。この後、該パケットの入力端子番号を、時刻Ｔhにおけるスイッチ状態表の獲得した出力端子番号の保持ポートに書き込み（ステップＳ２７）、該パケットが通過するのに要するクロック数を保持クロックに書き込む（ステップＳ２８）。次に、保持クロックに書き込まれたクロック数を１つ減らし、現在時刻Ｔhを１つ進め（ステップＳ２９）、保持クロックに書き込まれたクロック数が０でないか否かを調べる（ステップＳ３０）。ステップＳ３０で、０でない場合は（ＹＥＳ）、ステップＳ２７に戻り、０の場合は（ＮＯ）、ステップＳ２２に戻る。

【００４９】

一方、ステップＳ２５で、取り出したパケットが調停に負けると（ＮＯ）、負けたパケットをパケット集合Ｕtsから取り除き、次の時刻のパケット集合Ｕts+1に加え、同じノードの後続パケットの発行の重複がなくなるまで１つずつパケット集合をずらし（ステップＳ３１）、ステップＳ２２に戻る。また、ステップＳ２２で、ポート要求待ち行列がない場合は（ＮＯ）、出力端子番号ＰＯcurを１つ進め（ステップＳ３２）、出力端子番号ＰＯcurがスイッチの出力端子数Ｎport未満であるか否かを調べる（ステップＳ３３）。ステップＳ３３で、出力端子数Ｎport未満である場合（ＹＥＳ）、ステップＳ２２に戻り、出力端子数Ｎport未満でない場合（ＮＯ）は、本フローは終了して図９のステップＳ１４に進む。

【００５０】

上記のようなスケジューリング方法に対して、具体的な例を示しながら説明する。例えば、アクセス発行時刻Ｔsのパケットのスケジューリングにおいて、時刻１５０００のある階層ネットワークに属するスイッチが図１１で示したような状況である場合について説明する。なお、図１１では、エクスチェンジャを構成する５入力５出力のスイッチエレメントを例にして示している。

図１１において、出力端子＃２は、入力端子＃４によって２クロックだけ保持されている。入力端子のパケットは、ルーティングタグによって適切な出力端子のポート要求待ち行列に入る。図１１のスイッチ状態表は、スケジューリングが行われる前の状態を示しており、コンパイラは、図１１のスイッチ状態表を基にして調停を行い、図１２で示したスイッチ状態表を作成する。

【００５１】

図１１において、出線競合があるのは出力端子＃１であり、この場合、コンパイラは、パケットのヘッダ内の優先度に応じて調停を行う。仮に入力端子＃１のパケットが調停に勝ったとすると、調停に負けた入力端子＃０のパケットは、コンパイラによって、アクセス発行時刻Ｔsのパケット集合Ｕtsの要素から取り除かれ、次の発行時刻のパケット集合Ｕts+1に加えられ、図１２のように入力端子＃０のパケットが出力端子＃１に対するエントリから外される。調停に勝った入力端子＃１のパケットは、コンパイラによって、図１２のように出力端子＃１の保持ポートに入れられると共に出力端子＃１の保持クロックに１が書き込まれ、更に出力端子＃１の状態がホールドに設定される。

【００５２】

次に、出力端子＃２においては、図１１のようにすでに状態がホールドに設定され入力端子＃４のパケットによって２クロック保持されている。このため、出力端子＃２を要求している入力端子＃３のパケットは、調停に負けたパケットと同様に、コンパイラによって、パケット集合Ｕtsの要素から取り除かれて、次の発行時刻のパケット集合Ｕts+1に加えられ、図１２のように入力端子＃３のパケットが出力端子＃２に対するエントリから外される。出力端子＃４においては、図１１のように状態がリリースに設定されており競合する入力端子のパケットもないことから、入力端子＃２のパケットは、コンパイラによって、図１２のように出力端子＃４の保持ポートに入れられると共に出力端子＃４の保持クロックに１が書き込まれ、更に出力端子＃４の状態がホールドに設定される。

【００５３】

このように、調停が完了して図１２のスイッチ状態表が作成されると、コンパイラによって、現在時刻が１つ進められ、出力端子を獲得したパケットは、該出力端子に接続されるスイッチの入力端子にエントリされ、出力端子＃４のパケットは、上位階層のスイッチの入力端子にエントリされる。該エントリされたパケットは、コンパイラによって、上記と同様にポート要求待ち行列にエントリし、調停及び出力端子の獲得操作を目的地に到着するまで繰り返される。なお、上記説明では、階層構造化する多段結合網としてクロス網を使用した場合を例にして示した。しかし、本発明は、これに限定するものではなく、オメガ(Omega)網、

ベースライン(Baseline)網、デルタ(Delta)網及び「Generalized Cube」等の一般的な多段結合網を使用して階層構造化を行うようにしても実現することができる。

【００５４】

ここで、上記スケジューリング方法では、調停に負けたパケットは次以降の時刻のパケット集合に加えられるようにした。これに対して、１つのクロス網内でパケット転送を行う場合は、レベル１のエクスチェンジャのスイッチエレメントに対するスケジューリングを行った際、調停に負けたパケットをレベル１のエクスチェンジャの他のスイッチエレメントにおける空きのスイッチを使用して転送するようにしてもよい。このようにする場合のスケジューリング方法について、図１３の１つのクロス網、すなわちクラスタＤ０を例にして説明する。

【００５５】

クロス網の性質上、クロス網内の目的地への経路は、２段目のエクスチェンジャと３段目のコンセントレータによって決定されることから、１段目のディストリビュータは任意の出力を選んでよい。クロス網の転送性能は、２段目のエクスチェンジャのスケジューリング性能に大きく依存するため、２段目のエクスチェンジャのスケジューリング結果に応じた出力端子に転送する。したがって、２段目のエクスチェンジャのスケジューリングを先に行ってから、１段目のディストリビュータのスケジューリングを行う。

【００５６】

クロス網の転送性能は、２段目のエクスチェンジャのスケジューリング性能に左右されるため、非常に重要である。２段目のエクスチェンジャを効率よく生かすために上記のようなスイッチ状態表の他に、クラスタ別アクセスリストＡＬ(Access List)とクラスタ別空きポートカウンタＶＰＣ(Varid Port Counter)をスケジューリングで使用する。クラスタ別アクセスリスト（以下、アクセスリストと呼ぶ）ＡＬとは、各レベル０のクラスタから出力されたパケットが、どのレベル０のクラスタに向かっているかを記録したリストであり、クラスタ別空きポートカウンタ（以下、空きポートカウンタと呼ぶ）ＶＰＣは、各レベル０のクラスタごとに、接続されている出力端子がいくつあるかを表すカウンタである。

【００５７】

ここで、コンパイラによって行われるアクセスリストＡＬ及び空きポートカウンタＶＰＣの作成方法について、図１３を用いて説明する。なお、図１３においても、４を基数、すなわち４つのプロセッサエレメントＰＥ０～ＰＥ３を有する４つのクラスタＡ０～Ａ３を備える４つのクラスタＤ０～Ｄ３で構成された階層構造クラスタリングを実現した場合を例にして示している。クラスタＤ０は、クラスタＡ０～Ａ３及びエクスチェンジャをなすスイッチエレメントＳＥ０～ＳＥ３で構成されている。更に、クラスタＡ０～Ａ３は、対応するスイッチエレメントＳＤ０～ＳＤ３及びＳＣ０～ＳＣ３とプロセッサエレメントＰＥ０～ＰＥ３からそれぞれ形成されている。

【００５８】

このような構成において、まず、アクセスリストＡＬの作成方法について説明する。

コンパイラは、スイッチエレメントＳＤ０からスイッチエレメントＳＥ０～ＳＥ３に転送されたすべてのパケットのヘッダを調べ、該各パケットにおけるあて先のクラスタ番号をアクセスリストＡＬにそれぞれ書き込む。例えば、スイッチエレメントＳＤ０からの各パケットは、スイッチエレメントＳＥ０～ＳＥ３のルーティングタグとして、クラスタＡ１及びＡ３の２つの要素を有していた場合、アクセスリストＡＬにおけるクラスタＡ０には、クラスタ番号Ａ１及びＡ３が書き込まれる。

【００５９】

次に、空きポートカウンタＶＰＣの作成方法について説明する。

コンパイラは、作成したアクセスリストＡＬに基づいて、クラスタＡ０～Ａ３ごとに、割り当てられる出力端子の空きがいくつあるかを表すカウント値ＣＴ０～ＣＴ３を下記（３）式から対応させて算出する。

ＣＴｇ＝(２段目のスイッチ数)－(クラスタ別アクセスリストの要素数)……（３）

なお、上記（３）式において、ｇ＝０～３である。

例えば、クラスタＡ０に対するカウント値ＣＴ０は、ＣＴ０＝４－２＝２とな

る。

【００６０】

次に、アクセスリストＡＬ及び空きポートカウンタＶＰＣを用いてコンパイラによって行われるスケジューリングアルゴリズムについて説明する。ただし、現在のアクセスリストをＡＬcurとし、調停後のアクセスリストをＡＬnewとする。

まず、コンパイラは、現在のアクセスリストＡＬcurにおける要素数が最も少ないクラスタのパケットから順に、スイッチエレメントＳＥ０から順に優先的に割り当てていく。次に、コンパイラは、アクセスリストＡＬcurにおける１つのクラスタの要素（あて先クラスタ番号）において、パケットの送り主とあて先が同じクラスタである要素の優先度を最も低くし、同じでない場合は、例えばあて先のクラスタ番号の小さい方から順に割り当てる。なお、あて先のクラスタ番号の大きい方から順に割り当てるようにしてもよい。

【００６１】

また、コンパイラは、アクセスリストＡＬcurの要素に対応する空きポートカウンタが０である場合、必然的にスケジューリング不能となり、該要素を現在の時刻のパケット集合から取り除き、次の時刻のパケット集合に加える。このようなパケットの割り当てで競合が発生した場合、コンパイラは、パケットの優先度又はラウンドロビン方式等で調停を行う。該調停に勝ったパケットは、コンパイラによって、アクセスリストＡＬcurの要素から取り除かれ、対応する空きポートカウンタＶＰＣにおける端子数を示すカウント値をデクリメントする。この後、コンパイラは、対応する出力端子のスイッチ状態表に調停に勝ったパケットの入力端子番号を記入し、調停に勝ったクラスタに処理済みのチェックを入れる。

【００６２】

次に、コンパイラは、調停に負けたパケットと、調停に勝ったパケットと同じあて先のクラスタを指定しているパケットとを、現在時刻のアクセスリストＡＬcurから取り除き、次の時刻のアクセスリストＡＬnewに移動させる。コンパイラは、このような処理を、アクセスリストＡＬcurのすべてのクラスタＡ０～Ａ３に対して行って処理済みチェックを入れるまで行う。アクセスリストＡＬcurにおけるすべてのクラスタＡ０～Ａ３に処理済みチェックを入れると、コンパイラ

は、アクセスリストＡＬcurの全要素をアクセスリストＡＬnewに移し、該アクセスリストＡＬnewをＡＬcurとして上記一連の処理を各クラスタごとの要素がすべてなくなるまで行う。

【００６３】

図１４は、アクセスリストＡＬ及び空きポートカウンタＶＰＣを使用したクロス網内のスケジューリング方法を示したフローチャートであり、図１４を用いてクロス網内のスケジューリング処理の流れについてもう少し詳細に説明する。なお、図１４の各フローで行われる処理は、特に明記しない限りコンパイラによって行われる。

図１４において、まず最初に、クロス網内のレベル０のクラスタでアクセスリストＡＬが空でないクラスタの集合ＵＣＬの内、最も要素の少ないクラスタを現在注目しているクラスタ番号ＣＬcurに設定する（ステップＳ４１）。なお、最も要素数が少ないクラスタが複数ある場合は、いずれか１つを選択してクラスタ番号ＣＬcurに設定する。

【００６４】

次に、クラスタ番号ＣＬcurのクラスタにおけるアクセスリストＡＬcurの要素であるパケットを１つ選択し（ステップＳ４２）、該選択したパケットにおけるあて先クラスタの空きポートカウンタＶＰＣが０であるか否かを調べる（ステップＳ４３）。ステップＳ４３で、空きポートカウンタＶＰＣが０である場合（ＹＥＳ）、選択したパケットを、パケット集合Ｕtsとアクセスリストから取り除き、次の時刻のパケット集合Ｕts+1にずらし、時刻Ｔs以降の時刻に発行される該パケットは、重複がなくなるまで後の時刻のパケット集合にずらされ（ステップＳ４４）、ステップＳ４２に戻る。

【００６５】

また、ステップＳ４３で、空きポートカウンタＶＰＣが０でない場合（ＮＯ）、選択したパケットを、空いている出力端子があるスイッチエレメントＳＥ０～ＳＥ３の内、最も番号の小さいスイッチエレメントの出力端子に割り当てる（ステップＳ４５）。次に、クラスタ集合ＵＣＬにおいて、アクセスリストＡＬcurで競合したパケットを有するクラスタの有無を調べ（ステップＳ４６）、競合し

たパケットを有するクラスタがある場合（ＹＥＳ）、該競合したパケットをアクセスリストＡＬcurから取り除いてアクセスリストＡＬnewに加え（ステップＳ４７）、ステップＳ４６に戻る。

【００６６】

また、ステップＳ４６で、競合したパケットを有するクラスタがない場合（ＮＯ）、対応する空きポートカウンタＶＰＣのカウント値を１つ減らし、現在注目しているクラスタ番号ＣＬcurをクラスタ集合ＵＣＬから削除する（ステップＳ４８）。次に、クラスタ集合ＵＣＬが空集合でないか否かを調べ（ステップＳ４９）、空集合でない場合は（ＹＥＳ）、ステップＳ４１に戻る。また、ステップＳ４９で、空集合の場合は（ＮＯ）、アクセスリストＡＬcurの全パケットをアクセスリストＡＬnewに移して該アクセスリストＡＬnewをＡＬcurとし、アクセスリストが空でないクラスタをクラスタ集合ＵＣＬの要素とする（ステップＳ５０）。次に、クラスタ集合ＵＣＬが空集合か否かを調べ（ステップＳ５１）、空集合の場合は（ＹＥＳ）、本フローは終了し、空集合でない場合は（ＮＯ）、ステップＳ４１に戻る。

【００６７】

このようなコンパイラによる処理を、具体的な例を用いて説明する。図１５は、アクセスリストＡＬcurの初期状態の例を示した図であり、図１６は、空きポートカウンタＶＰＣの初期状態の例を示した図である。図１５及び図１６で示した場合を例にして説明する。

まず、コンパイラは、アクセスリストＡＬcurにおける要素数の最も少ないクラスタＡ１におけるあて先がクラスタＡ２であるパケットを処理し、アクセスリストＡＬcurから削除する。更に、コンパイラは、レベル１のエクスチェンジャをなすスイッチエレメントＳＥ０の出力端子＃２を確保してスイッチ状態表に記録する。

【００６８】

次に、コンパイラは、空きポートカウンタＶＰＣにおける出力端子＃２のカウント値を１減らし、アクセスリストＡＬcurのクラスタＡ１に処理済みのチェックを入れる。一方、コンパイラは、空きポートカウンタＶＰＣにおける出力端子

＃２のカウンタ値が０であることから、アクセスリストＡＬcurにおいて、クラスタＡ３におけるあて先がクラスタＡ２であるパケットを削除して、再転送するために次の時刻のアクセスリストＡＬnewに移す。

【００６９】

次に、コンパイラは、アクセスリストＡＬcurにおいて、要素数が次に少ないクラスタＡ３におけるあて先がクラスタＡ０であるパケットを、スイッチエレメントＳＥ０の出力端子＃０を確保してスイッチ状態表に記録する。更に、コンパイラは、空きポートカウンタＶＰＣの出力端子＃０のカウント値を１減らし、アクセスリストＡＬcurのクラスタＡ３に処理済みのチェックを入れる。

【００７０】

同様に、コンパイラは、アクセスリストＡＬcurにおいて、クラスタＡ２におけるあて先がクラスタＡ０であるパケットを削除して、再転送するために次の時刻のアクセスリストＡＬnewに移す。この時点で、クラスタＡ０及びＡ２の要素数が共に２つとなることから、コンパイラは、調停を行ってクラスタＡ０を先に処理するものとする。コンパイラは、アクセスリストＡＬcurにおけるクラスタＡ０のあて先がクラスタＡ１であるパケットを、スイッチエレメントＳＥ０の出力端子＃１に割り当ててスイッチ状態表に記録し、アクセスリストＡＬcurから削除する。

【００７１】

更に、コンパイラは、空きポートカウンタＶＰＣにおける出力端子＃１のカウント値を１減らし、アクセスリストＡＬcurのクラスタＡ０に処理済みのチェックを入れる。この後、コンパイラは、アクセスリストＡＬcurにおいて、クラスタＡ２におけるあて先がクラスタＡ１であるパケットを削除して、再転送するために次の時刻のアクセスリストＡＬnewに移す。最後に、コンパイラは、処理済みのチェックが入っていないクラスタＡ２におけるあて先がクラスタＡ３であるパケットを同様にして処理する。

【００７２】

更に、コンパイラは、空きポートカウンタＶＰＣの出力端子＃３のカウンタ値を１減らし、アクセスリストＡＬcurのクラスタＡ２に処理済みのチェックを入

れてアクセスリストＡＬcurに対する処理が終了する。図１７は、クラスタＡ０～Ａ３の各パケットを１つずつ処理した後のアクセスリストＡＬnewを示し、図１８は、クラスタＡ０～Ａ３の各パケットを１つずつ処理した後の空きポートカウンタＶＰＣを示している。

【００７３】

次に、コンパイラは、新たなアクセスリストＡＬcurに対して、上記と同様の処理を行う。この際、パケットは、コンパイラによって、スイッチエレメントＳＥ１の出力端子に割り当てられる点が異なる。このため、アクセスリストＡＬnewがＡＬcurになるごとに、コンパイラによって割り当てられるスイッチエレメントが１つずつずれていくことになる。最終的に、コンパイラによってスケジューリングが施されたパケットの経路は、図１９の矢印のようになる。

【００７４】

なお、上記コンパイラによるスケジューリングでは、１つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合に調停が行われ、出力端子を確保することができなかったパケットは次の時刻のスイッチ状態表で該出力端子を要求する場合を例にして説明したが、これは一例であり、出力端子を確保することができなかったパケットが、その前の時刻のスイッチ状態表といったように他の時刻のスイッチ状態表で所望の出力端子を要求するようにしてもよい。

【００７５】

このように、本第１の実施の形態におけるマルチプロセッサシステム装置は、各プロセッサエレメント間を、階層構造を有する多段結合網で接続し、該多段結合網を構成する各スイッチに対して、あらかじめコンパイラによって静的にスケジューリングを行い、階層構造を有する多段結合網を無衝突でエミュレーションするようにした。このことから、パケットの衝突時に動的に行っていたパケットの待ち合わせを、すべてコンパイル時に管理することができるため、パケットの動的な待ち合わせに必要なＦＩＦＯ等のハードウェアを大幅に削減することができプロセッサ間での無同期実行を行うためのネットワーク環境を整えることができる。更に、マルチプロセッサシステム装置において無同期実行を可能にすることができるため、同期させるためのハードウェアのオーバヘッドを削減すること

ができ、並列処理の効率を向上させることができる。

【００７６】

また、階層構造を有する多段結合網の基本網にクロス網を使用して１つのクロス網内でパケット転送を行う場合、レベル１のエクスチェンジャのスイッチエレメントに対するスケジューリングを行った際、調停に負けたパケットをレベル１のエクスチェンジャの他のスイッチエレメントにおける空きのスイッチを使用して転送するようにしてもよい。このようにすることによって、パケット転送効率を向上させることができる。

【００７７】

第２の実施の形態.

上記第１の実施の形態では、クロス網間を接続するための２段目のレベル１のエクスチェンジャ、すなわち図６における各クロス網内のそれぞれのスイッチエレメントＳＥ０～ＳＥ３にすべてのパケットが集中する構造であるため、ホットスポット（hot spot）が形成されて性能が著しく低下する場合があった。このことから、上位階層から下位階層へ配送するダウンストリーム用のスイッチとして、レベル１のコンセントレータを付加するようにしてもよく、このようにしたものを本発明の第２の実施の形態とする。なお、本第２の実施の形態におけるマルチプロセッサシステム装置の例を示した概略のブロック図、及びプロセッサエレメントの構成例を示した概略のブロック図は、図１及び図２と同様であるので省略する。

【００７８】

図２０及び図２１は、本発明の第２の実施の形態における階層構造クラスタリングを実現したマルチプロセッサシステム装置の例を示した図である。図２０は、各クロス網と拡張ネットワークとのアップリンク結合を示しており、図２１は、各クロス網と拡張ネットワークとのダウンリンク結合を示している。なお、図２０及び図２１では、図６と同じものは同じ符号で示しており、ここではその説明を省略すると共に図６との相違点のみ説明する。また、図２０及び図２１においても、４を基数、すなわち４つのプロセッサエレメントＰＥ０～ＰＥ３を有する４つのクラスタＡ０～Ａ３を備える４つのクラスタＤ０～Ｄ３で構成された階

層構造クラスタリングを実現した場合を例にして示し、プロセッサエレメントは省略して示している。

【００７９】

図２０及び図２１における図６との相違点は、レベル１のエクスチェンジャにおける、上位階層ネットワークへのパケット配送機能（アップストリーム）と、下位階層ネットワークへのパケット配送機能（ダウンストリーム）を２つに分け、上位階層から下位階層へパケットを配送するダウンストリーム用のスイッチとしてスイッチングエレメントＳＣｂ０～ＳＣｂ３からなるレベル１のコンセントレータを付加すると共に、下位階層から上位階層へパケットを配送するアップストリームは、第１の実施の形態と同様に各クラスタＤ０～Ｄ３におけるそれぞれのスイッチングエレメントＳＥ０～ＳＥ３からなるレベル１のエクスチェンジャによって行われる。

【００８０】

あるプロセッサエレメント内のプロセッサＰＵが、他のプロセッサエレメントＰＥのプロセッサＰＵとデータの授受を行う場合、相手のプロセッサエレメントＰＥのメモリＭＥにデータを書き込むことによって通信を行う。メモリＭＥにデータが書き込まれたプロセッサエレメントのプロセッサＰＵは、メモリＭＥに書き込まれたデータを読み出すことによりデータ通信が成立する。

【００８１】

以下、プロセッサエレメント間のデータ通信の流れについて図２２を用いて説明する。

図２２において、プロセッサエレメントＰＥａからプロセッサエレメントＰＥｂにデータ転送する場合を例にして説明する。まず、プロセッサエレメントＰＥａにおいて、プロセッサＰＵａからネットワークインタフェースＮＩａにアドレス及び転送データが送られる。

【００８２】

次に、ネットワークインタフェースＮＩａは、入力されたアドレスに基づいてパケットを生成し、階層構造の多段結合網ＭＩＮにパケットを投入する。該投入されたパケットは、階層構造の多段結合網ＭＩＮを介して、プロセッサエレメン

トＰＥｂのネットワークインタフェースＮＩｂに入力され、該ネットワークインタフェースＮＩｂは、入力されたパケットを解体してメモリＭＥｂに書き込む。プロセッサＰＵｂは、メモリＭＥｂに書き込まれたデータを読み出してデータ通信が完了する。

【００８３】

ここで、レベル１のエクスチェンジャに送り出されたパケットが、同一のクロス網内で処理されるパケット、すなわちデスティネーションが同一クロス網内である場合について図３を用いて説明する。

図３において、プロセッサエレメントからパケットが、１段目のディストリビュータに送り出され、該１段目のディストリビュータでスイッチングされて２段目のレベル１のエクスチェンジャに送られる。レベル１のエクスチェンジャに送り出されたパケットは、該レベル１のエクスチェンジャによって、最終段のレベル０のコンセントレータにスイッチングされて送り出される。

【００８４】

更に、レベル０のコンセントレータに送り出されたパケットは、レベル０のコンセントレータで適切にスイッチングされて、デスティネーションのプロセッサエレメントに送り出されて階層構造の多段結合網ＭＩＮでのデータ通信が完了する。なお、あて先のプロセッサエレメントに送り出されたパケットは、図２２で説明したように、プロセッサエレメントのメモリに格納される。

【００８５】

次に、レベル１のエクスチェンジャに送り出されたパケットが、他のクロス網で処理されるパケット、すなわちデスティネーションが他のクロス網である場合について図２３を用いて説明する。なお、図２３では、プロセッサエレメントＰＥａからプロセッサエレメントＰＥｂにデータ転送する場合を例にして説明する。

レベル１のエクスチェンジャをなすスイッチエレメントＳＥ１に送り出されたパケットは、該スイッチエレメントＳＥ１によって、拡張段への出力端子にスイッチングされ、同一レベルのクラスタ内に入るまで上位階層のエクスチェンジャ、この場合レベル２のエクスチェンジャをなすスイッチエレメントＳＥａ１に送

られる。

【００８６】

同一クラスタ内に入ったパケットは、適切な出力にスイッチングされて階層を下る。例えば図２３の場合、スイッチエレメントＳＥａ１に送られたパケットは、スイッチエレメントＳＥａ１によって、ダウンストリームのレベル１のコンセントレータをなすスイッチエレメントＳＣｂ１に転送される。スイッチエレメントＳＣｂ１に転送されたパケットは、スイッチエレメントＳＣｂ１で適切にスイッチングされて、デスティネーションのプロセッサエレメントであるプロセッサエレメントＰＥｂに送り出される。このようにして、階層構造の多段結合網ＭＩＮでのデータ通信が完了する。このような構成において、階層構造を有する多段結合網の静的スケジューリング方法は上記第１の実施の形態と同様であるのでその説明を省略する。

【００８７】

このように、本第２実施の形態におけるマルチプロセッサシステム装置は、上位階層から下位階層へ配送するダウンストリーム用のスイッチとして、スイッチングエレメントＳＣｂ０～ＳＣｂ３からなるレベル１のコンセントレータを付加し、下位階層から上位階層へパケットを配送するアップストリームは、スイッチングエレメントＳＥ０～ＳＥ３からなるレベル１のエクスチェンジャによって行うようにした。このことから、クロス網間を接続するためのレベル１のエクスチェンジャにすべてのパケットが集中しないようにしてホットスポットの形成を防止し、マルチプロセッサシステム装置の性能向上を図ることができる。

【００８８】

【発明の効果】

上記の説明から明らかなように、本発明のマルチプロセッサシステム装置によれば、各プロセッサエレメント及び多段結合網を、所定の数を基数とした階層構造にクラスタリングすると共に、各時刻ごとに生成された多段結合網における各スイッチの状態を示すスイッチ状態表を用いてあらかじめ静的にスケジューリングされたスケジュールに基づいて、プロセッサエレメント間のパケット転送を行うようにした。このことから、マルチプロセッサシステム装置において無同期実

行を可能にすることができるため、同期させるためのハードウェアのオーバヘッドを削減することができ、並列処理の効率を向上させることができる。

【００８９】

また、上記階層構造を有する多段結合網は、下位階層から上位階層へパケット転送を行うアップストリーム用の結合網と、上位階層から下位階層へパケット転送を行うダウンストリーム用の結合網とをそれぞれ備えるようにした。このことから、クロス網間を接続するためのエクスチェンジャをなす結合網にすべてのパケットが集中しないようにしてホットスポットの形成を防止し、マルチプロセッサシステム装置の性能向上を図ることができる。

【００９０】

具体的には、上記スイッチ状態表を、各スイッチごとの、出力端子を保持しているパケットの情報と、該出力端子を要求しているパケットの情報と、該出力端子の状態を示した情報とで構成した。このことから、各プロセッサエレメント及び多段結合網からなる大規模なシステムに対する静的なスケジューリングを容易に行うことができる。

【００９１】

また、多段結合網における１つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、他の時刻のスイッチ状態表で該出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うようにした。このことから、一般的な同時アクセスパターンに対して、無衝突のパケット転送を実現することができる。

【００９２】

一方、上記多段結合網がクロス網の場合、１つのクロス網内のパケット転送時に、多段結合網における１つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、パケットの要求がない他のスイッチの出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うようにしてもよい。このようにすることによって、パケット

転送効率を向上させることができ、マルチプロセッサシステム装置の性能向上を図ることができる。

【００９３】

具体的には、各パケットに対するスケジューリングをあらかじめコンパイラによって行うようにした。このことから、パケットの衝突時に動的に行っていたパケットの待ち合わせを、すべてコンパイル時に管理することができるため、パケットの動的な待ち合わせに必要なＦＩＦＯ等のハードウェアを大幅に削減することができプロセッサ間での無同期実行を行うためのネットワーク環境を整えることができる。

【図面の簡単な説明】

【図１】　本発明の第１の実施の形態におけるマルチプロセッサシステム装置の例を示した概略のブロック図である。

【図２】　プロセッサエレメントの構成例を示した概略のブロック図である。

【図３】　クロス網の例を示した図である。

【図４】　クロス網内におけるレベル０のネットワークの例を示した図である。

【図５】　クロス網内におけるレベル１のネットワークの例を示した図である。

【図６】　階層構造クラスタリングを実現したマルチプロセッサシステム装置の例を示した図である。

【図７】　スイッチ状態表の例を示した図である。

【図８】　スイッチ状態表を使用した静的スケジューリング方法を示したフローチャートである。

【図９】　スイッチ状態表を使用した静的スケジューリング方法を示したフローチャートである。

【図１０】　スイッチ状態表を使用した静的スケジューリング方法を示したフローチャートである。

【図１１】　調停を行う前のスイッチ状態表の例を示した図である。

【図１２】　調停を行った後のスイッチ状態表の例を示した図である。

【図１３】　クロス網の例を示した図である。

【図１４】　アクセスリストＡＬ及び空きポートカウンタＶＰＣを使用したクロス網内のスケジューリング方法を示したフローチャートである。

【図１５】　アクセスリストＡＬcurの初期状態の例を示した図である。

【図１６】　空きポートカウンタＶＰＣの初期状態の例を示した図である。

【図１７】　各パケットを１つずつ処理した後のアクセスリストＡＬnewの例を示した図である。

【図１８】　各パケットを１つずつ処理した後の空きポートカウンタＶＰＣの例を示した図である。

【図１９】　スケジューリング後の各パケットの経路を示した図である。

【図２０】　本発明の第２の実施の形態における階層構造クラスタリングを実現したマルチプロセッサシステム装置の例を示した図である。

【図２１】　図２０のマルチプロセッサシステム装置における各クロス網と拡張ネットワークとのダウンリンク結合例を示した図である。

【図２２】　プロセッサエレメント間のデータ通信の流れの例を示した図である。

【図２３】　図２０のマルチプロセッサシステム装置１ａにおけるプロセッサエレメント間のパケット転送の流れの例を示した図である。

【符号の説明】

１，１ａ　マルチプロセッサシステム装置

ＰＥ，ＰＥ０～ＰＥｍ　プロセッサエレメント

Ａ０～Ａｎ　レベル０のクラスタ

Ｂ０～Ｂｎ　レベル０の結合網

Ｃ０～Ｃｘ　レベル１の結合網

Ｄ０～Ｄｘ　レベル１のクラスタ

Ｅ０　レベル２の結合網

ＰＵ，ＰＵａ，ＰＵｂ　プロセッサ

ＭＥ，ＭＥａ，ＭＥｂ　メモリ

ＮＩ，ＮＩａ，ＮＩｂ　ネットワークインタフェース

ＳＣ０～ＳＣ３　コンセントレータ（レベル０）のスイッチエレメント

ＳＤ０～ＳＤ３　ディストリビュータ（レベル０）のスイッチエレメント

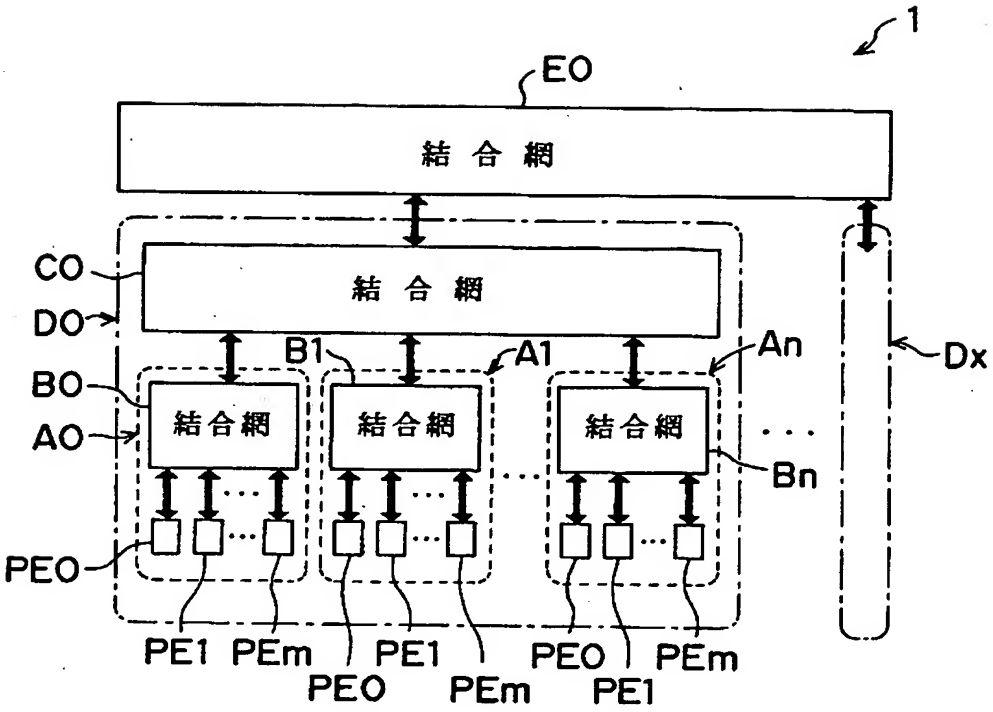
ＳＥ０～ＳＥ３　エクスチェンジャ（レベル０，１）のスイッチエレメント

ＳＥａ０～ＳＥａ３　エクスチェンジャ（レベル２）のスイッチエレメント

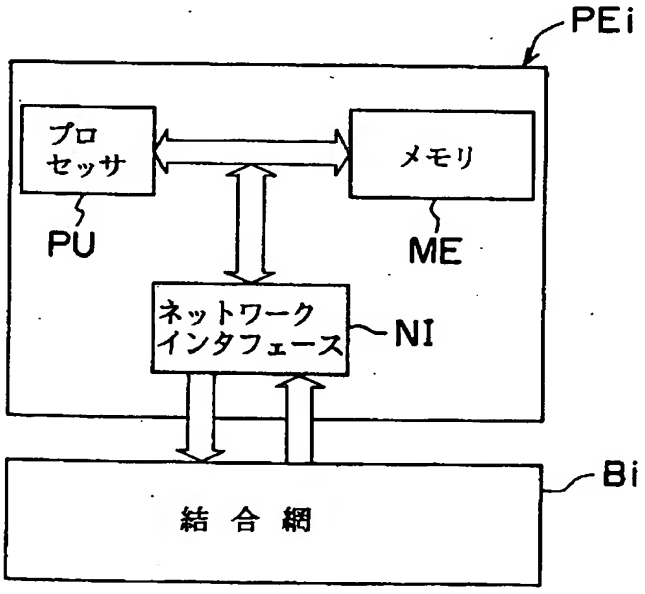
ＳＣｂ０～ＳＣｂ３　コンセントレータ（レベル１）のスイッチエレメント

【書類名】　図面

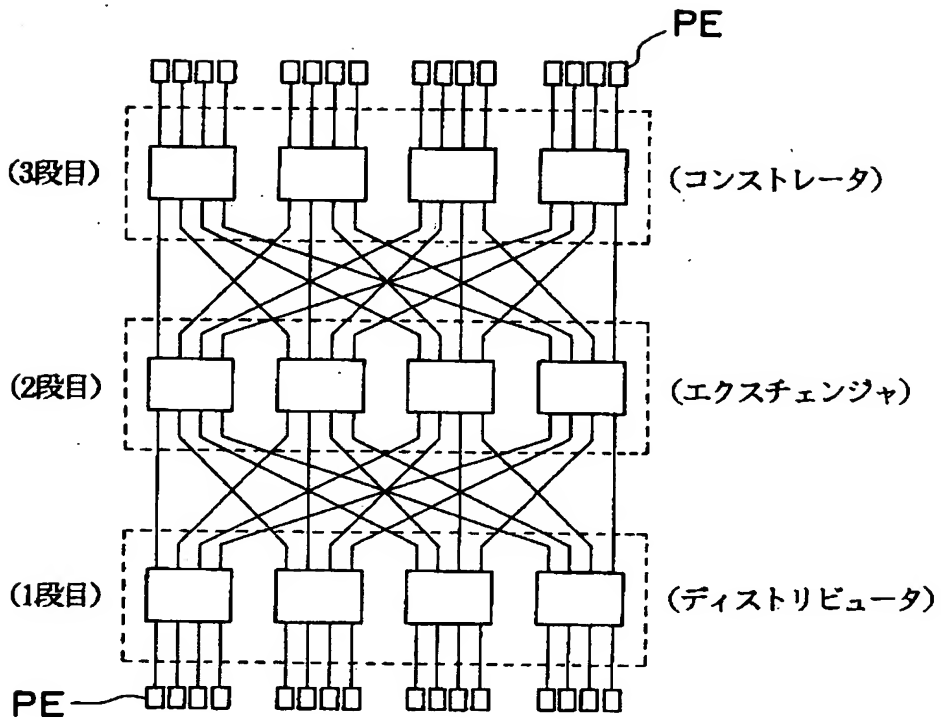
【図１】


1
EO
結 合 網
CO
DO
結 合 網
B1
A1
An
Dx
BO
AO
結合網
結合網
結合網
Bn
PEO
PE1
PEm
PE1
PEO
PEm
PEO
PEm
PE1

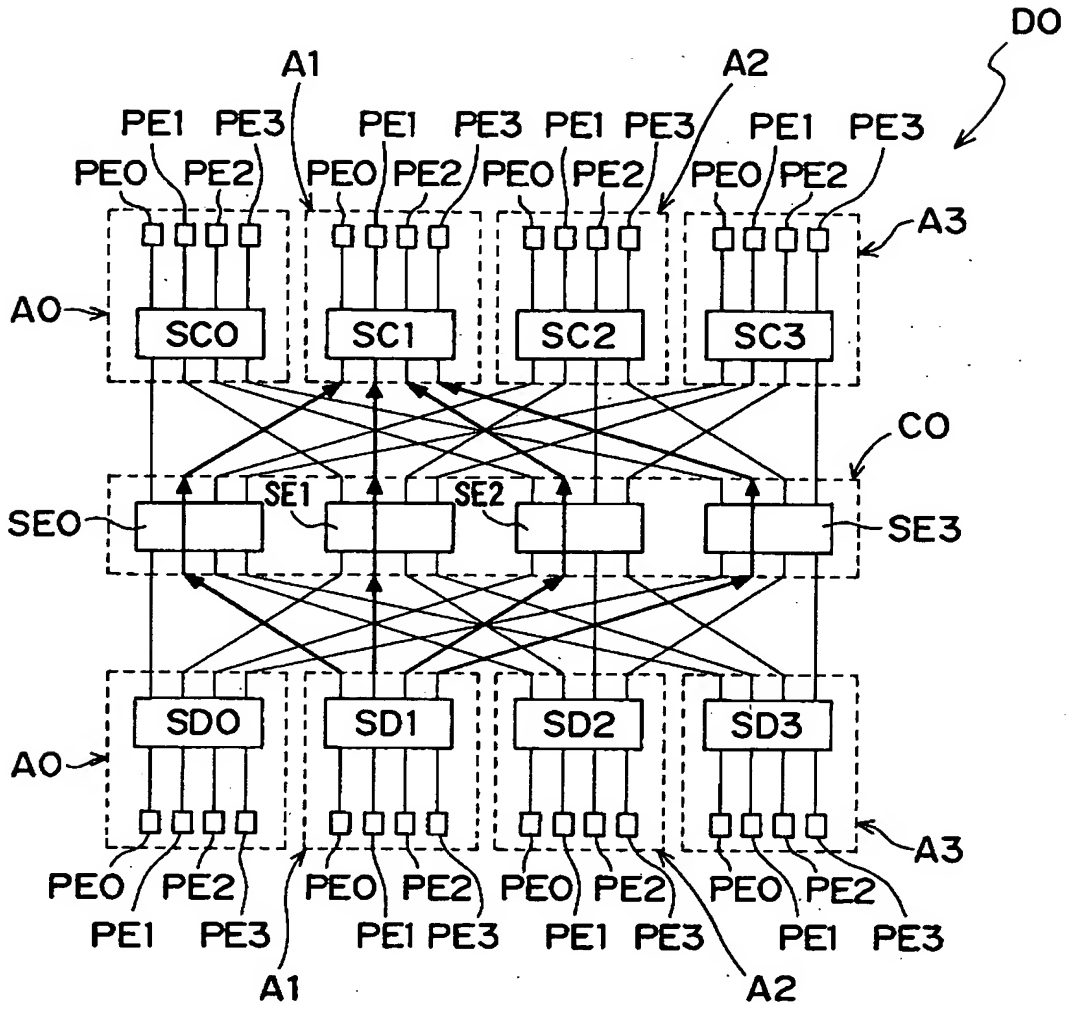
【図２】


PEi
プロセッサ
メモリ
PU
ME
ネットワークインタフェース
NI
Bi
結 合 網

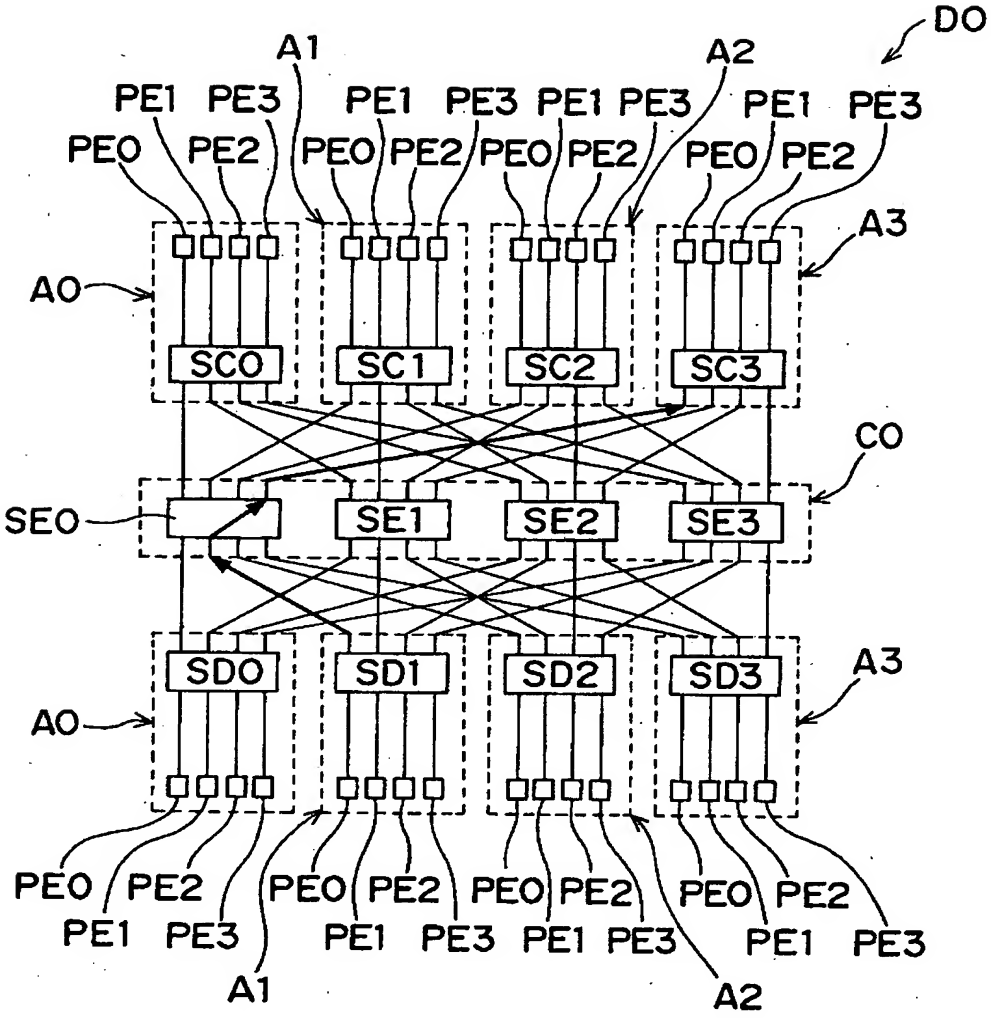
【図３】



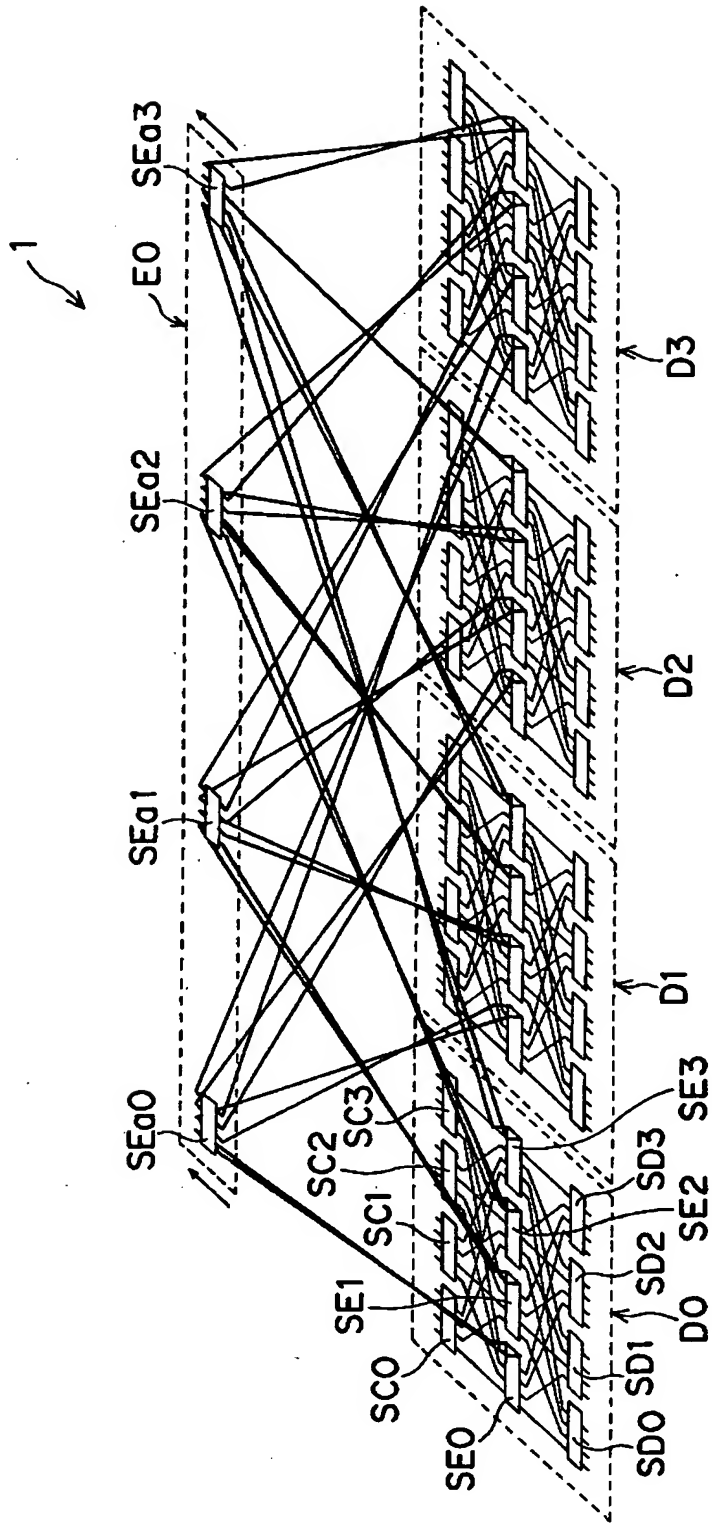
【図４】


DO
A1
A2
PE1 PE3 PE1 PE3 PE1 PE3 PE1 PE3
PE0 PE2 PE0 PE2 PE0 PE2 PE0 PE2
A3
A0
SC0
SC1
SC2
SC3
CO
SE1
SE2
SE0
SE3
SD0
SD1
SD2
SD3
A0
A3
PE0 PE2 PE0 PE2 PE0 PE2 PE0 PE2
PE1 PE3 PE1 PE3 PE1 PE3 PE1 PE3
A1
A2

【図５】


DO
A1
A2
PE1 PE3 PE1 PE3 PE1 PE3 PE1 PE3
PE0 PE2 PE0 PE2 PE0 PE2 PE0 PE2
A3
A0
SC0 SC1 SC2 SC3
CO
SE0 SE1 SE2 SE3
SD0 SD1 SD2 SD3
A3
A0
PE0 PE2 PE0 PE2 PE0 PE2 PE0 PE2
PE1 PE3 PE1 PE3 PE1 PE3 PE1 PE3
A1
A2

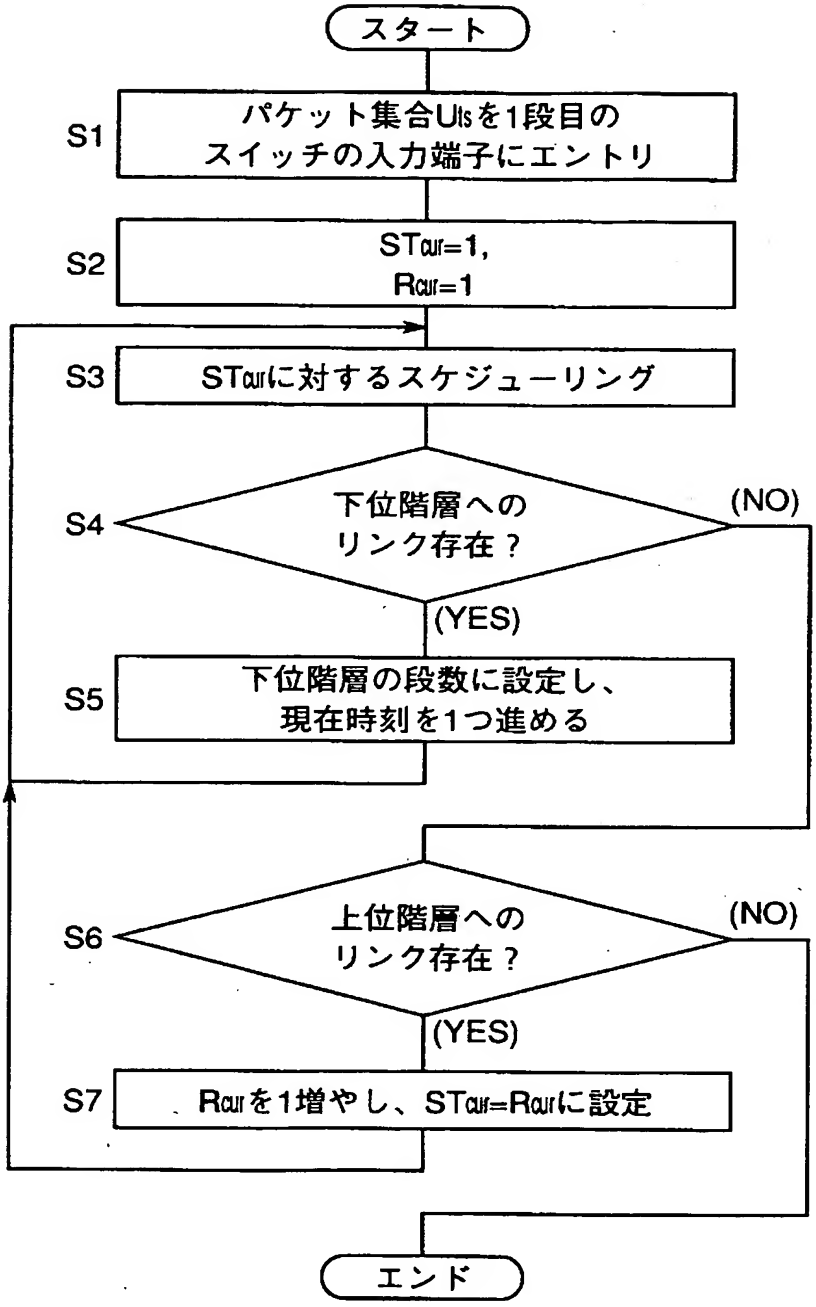
【図６】


1
E0
SEa3
SEa2
SEa1
SEa0
D3
D2
D1
D0
SE3
SE2
SE1
SE0
SD3
SD2
SD1
SD0
SC3
SC2
SC1
SC0

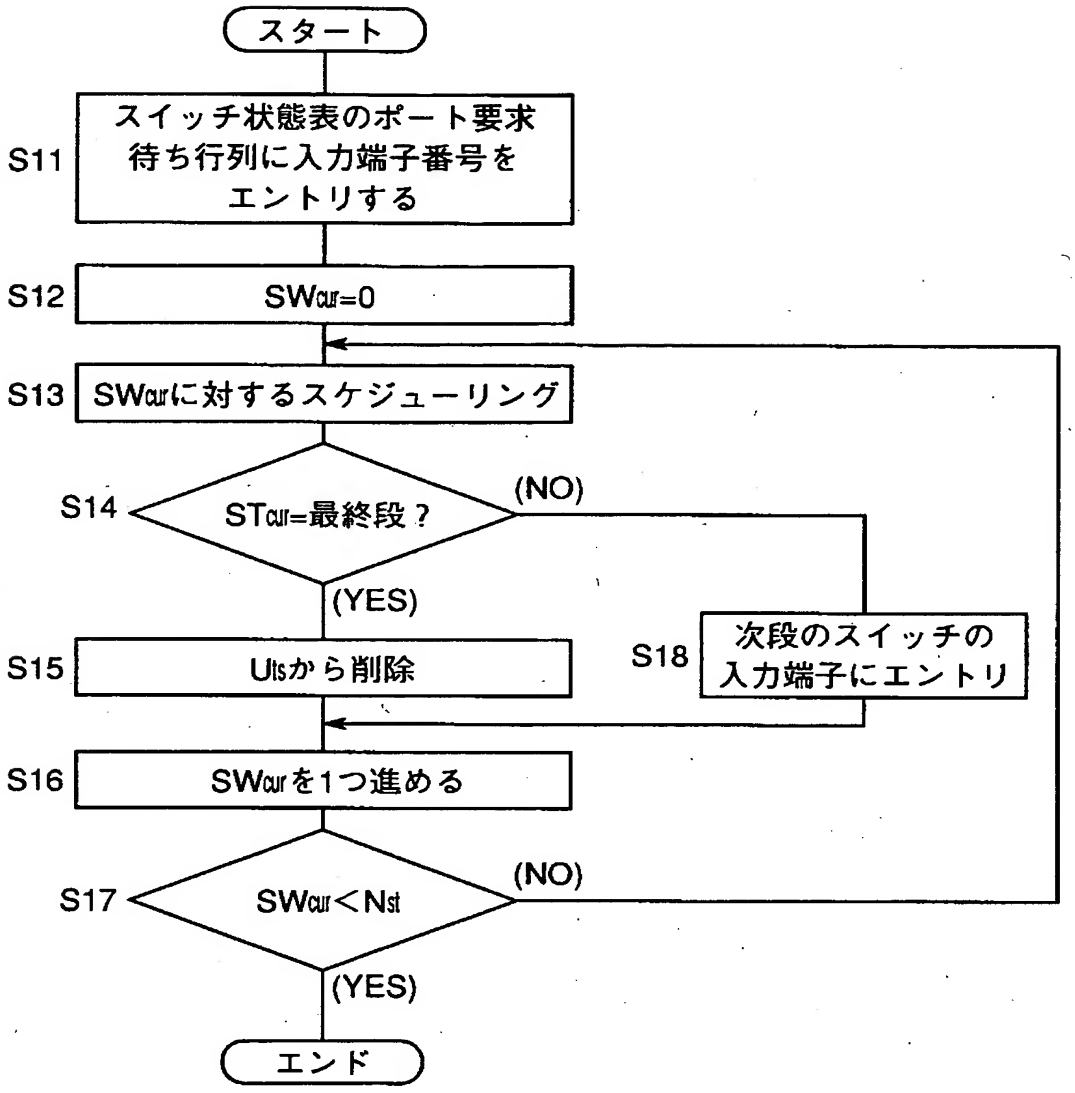
【図７】

出力端子番号	現在時刻	保持ポート	保持クロック	ポート要求待ち行列	状態
＃0	157843	＃3	2	－	ホールド
＃1	157843	－	0	＃0,＃2	リリース
＃2	157843	－	0	＃1	リリース
＃3	157843	－	0	－	リリース

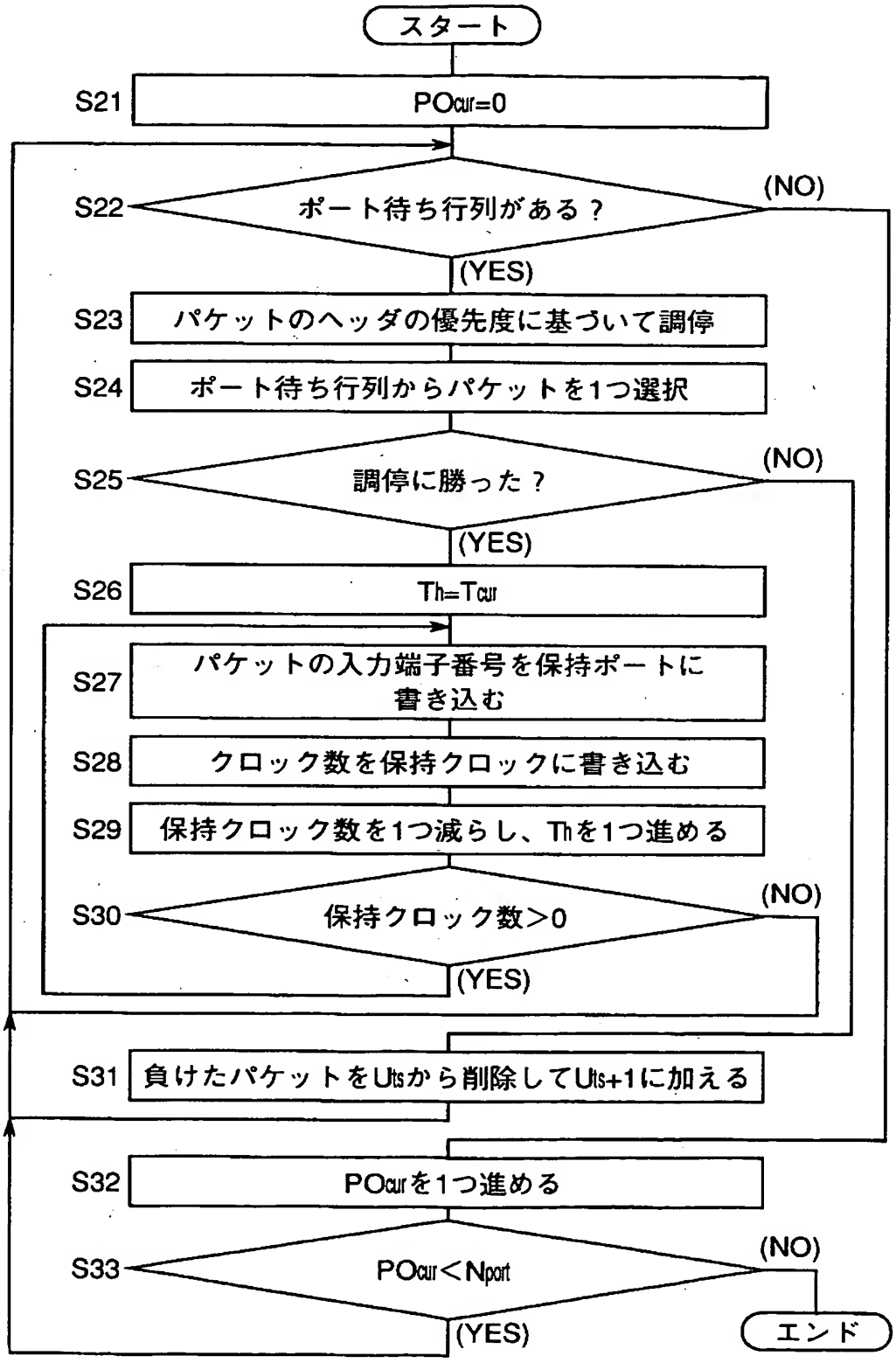
【図８】


スタート
S1
パケット集合Utsを1段目の
スイッチの入力端子にエントリ
S2
STcur=1,
Rcur=1
S3
STcurに対するスケジューリング
S4
下位階層への
リンク存在？
(NO)
(YES)
S5
下位階層の段数に設定し、
現在時刻を1つ進める
S6
上位階層への
リンク存在？
(NO)
(YES)
S7
Rcurを1増やし、STcur=Rcurに設定
エンド

【図９】


スタート
S11
スイッチ状態表のポート要求
待ち行列に入力端子番号を
エントリする
S12
SWcur=0
S13
SWcurに対するスケジューリング
S14
STcur=最終段？
(NO)
(YES)
S15
Utsから削除
S18
次段のスイッチの
入力端子にエントリ
S16
SWcurを1つ進める
S17
SWcur＜Nst
(NO)
(YES)
エンド

【図１０】


スタート
S21
POcur=0
S22
ポート待ち行列がある？
(NO)
(YES)
S23
パケットのヘッダの優先度に基づいて調停
S24
ポート待ち行列からパケットを1つ選択
S25
調停に勝った？
(NO)
(YES)
S26
Th=Tcur
S27
パケットの入力端子番号を保持ポートに書き込む
S28
クロック数を保持クロックに書き込む
S29
保持クロック数を1つ減らし、Thを1つ進める
S30
保持クロック数＞0
(NO)
(YES)
S31
負けたパケットをUtsから削除してUts+1に加える
S32
POcurを1つ進める
S33
POcur＜Nport
(NO)
(YES)
エンド

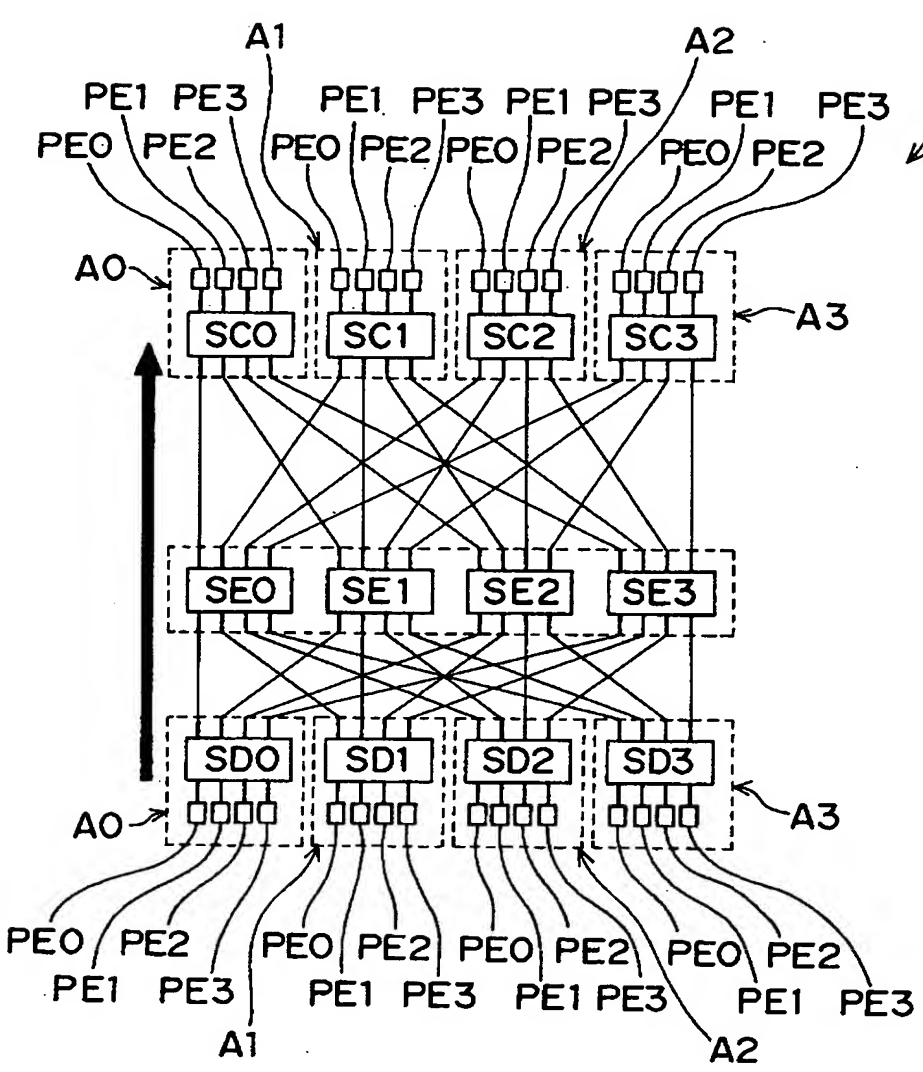
【図１１】

出力端子番号	現在時刻	保持ポート	保持クロック	ポート要求待ち行列	状態
＃0	15000	－	0	－	リリース
＃1	15000	－	0	＃0,＃1	リリース
＃2	15000	＃4	2	＃3	ホールド
＃3	15000	－	0	－	リリース
＃4	15000	－	0	＃2	リリース

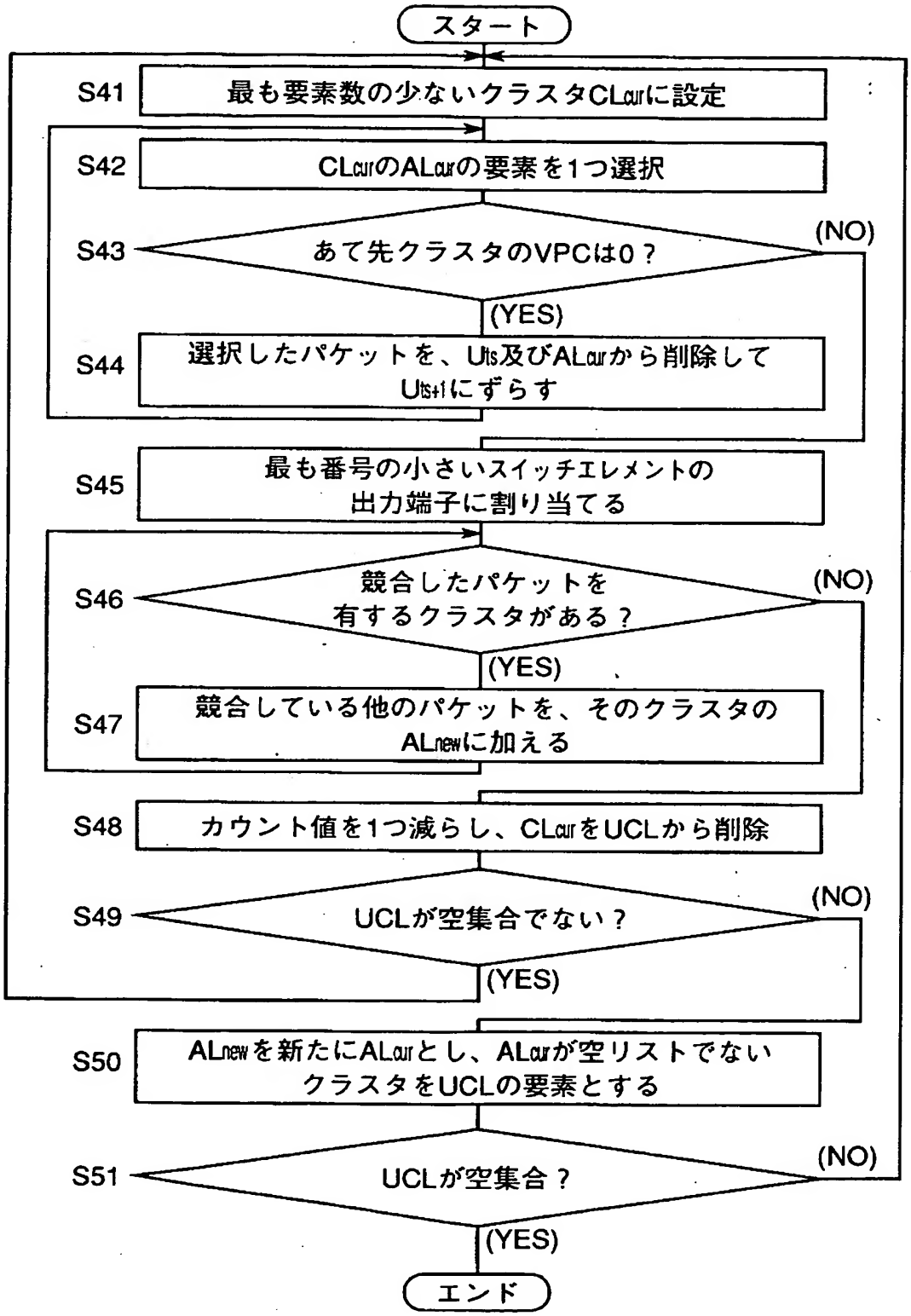
【図１２】

出力端子番号	現在時刻	保持ポート	保持クロック	ポート要求待ち行列	状態
＃0	15000	－	0	－	リリース
＃1	15000	＃1	1	－	ホールド
＃2	15000	＃4	2	－	ホールド
＃3	15000	－	0	－	リリース
＃4	15000	＃2	1	－	ホールド

【図１３】


A1
A2
DO
PE1 PE3 PE1 PE3 PE1 PE3 PE1 PE3
PE0 PE2 PE0 PE2 PE0 PE2 PE0 PE2
A0
SC0
SC1
SC2
SC3
A3
SE0
SE1
SE2
SE3
SD0
SD1
SD2
SD3
A0
A3
PE0 PE2 PE0 PE2 PE0 PE2 PE0 PE2
PE1 PE3 PE1 PE3 PE1 PE3 PE1 PE3
A1
A2

【図１４】


スタート
S41
最も要素数の少ないクラスタCLcurに設定
S42
CLcurのALcurの要素を1つ選択
S43
あて先クラスタのVPCは0？
(NO)
(YES)
S44
選択したパケットを、Uts及びALcurから削除して
Uts+1にずらす
S45
最も番号の小さいスイッチエレメントの
出力端子に割り当てる
S46
競合したパケットを
有するクラスタがある？
(NO)
(YES)
S47
競合している他のパケットを、そのクラスタの
ALnewに加える
S48
カウント値を1つ減らし、CLcurをUCLから削除
S49
UCLが空集合でない？
(NO)
(YES)
S50
ALnewを新たにALcurとし、ALcurが空リストでない
クラスタをUCLの要素とする
S51
UCLが空集合？
(NO)
(YES)
エンド

【図１５】

クラスタ番号	あて先のクラスタ番号
A0	A1,A3
A1	A2
A2	A0,A1,A3
A3	A0,A2

【図１６】

出力端子番号	#0	#1	#2	#3
カウント値	2	3	1	2

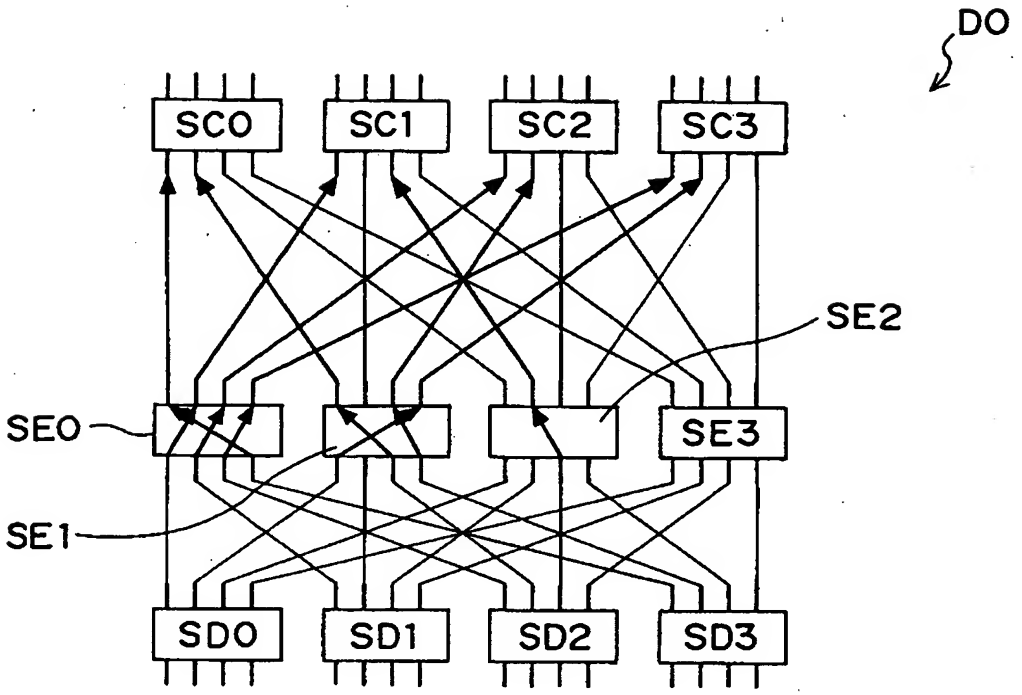
【図１７】

クラスタ番号	あて先のクラスタ番号
A0	A3
A1	—
A2	A0,A1
A3	A2

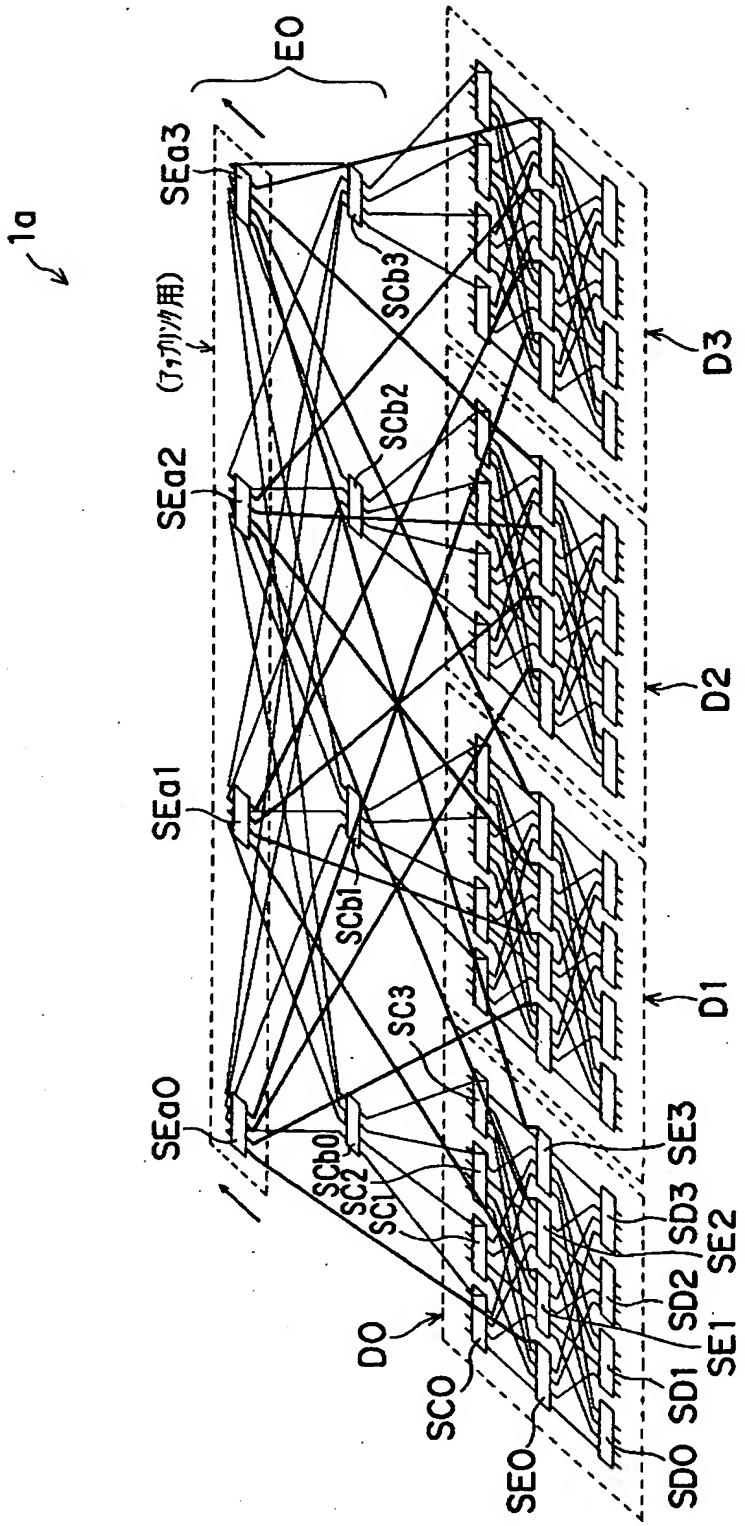
【図１８】

出力端子番号	#0	#1	#2	#3
カウント値	1	2	0	1

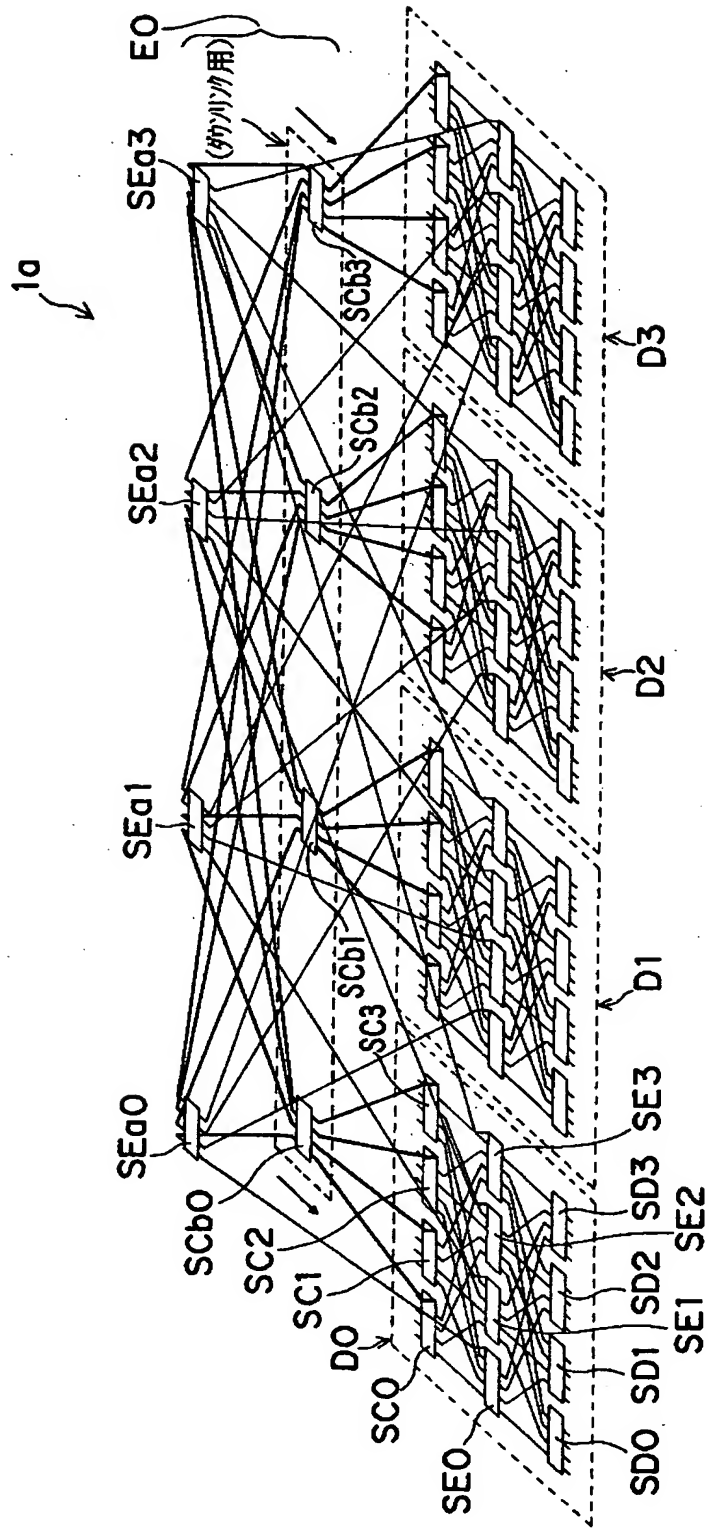
【図１９】



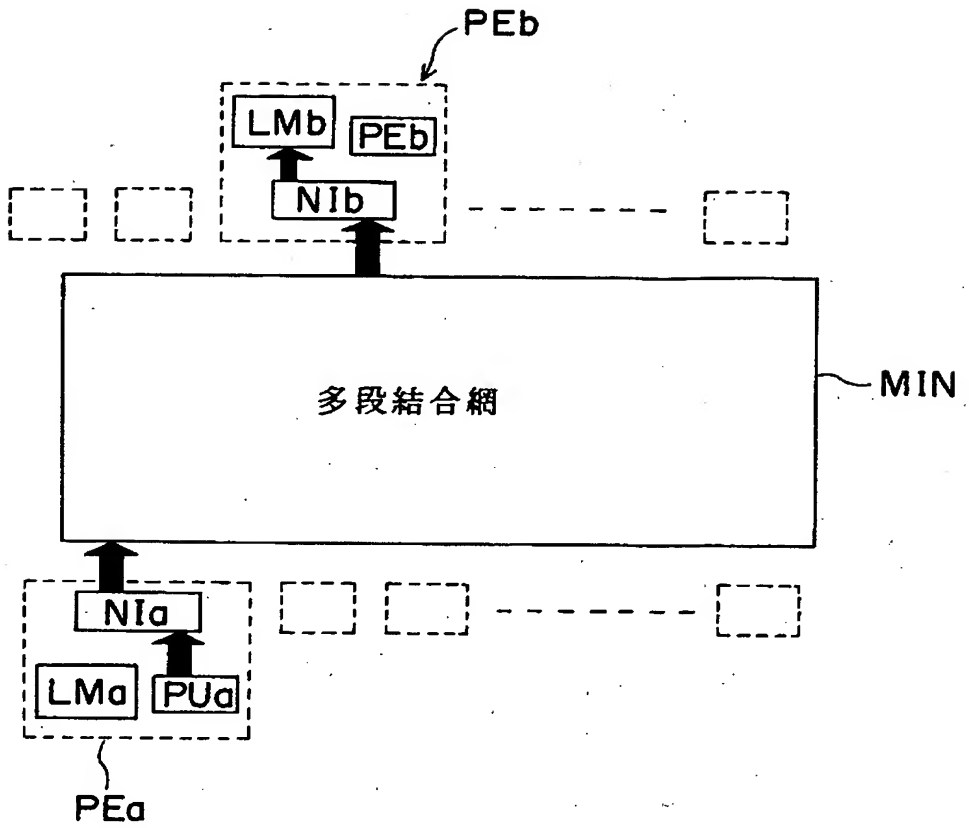
【図２０】


1a
EO
SEa3
SEa2
SEa1
SEaO
(テスト用)
SCb3
SCb2
SCb1
SCbO
SC3
SC2
SC1
SCO
SEO
DO
SDO
SD1
SD2
SD3
SE1
SE2
SE3
D1
D2
D3

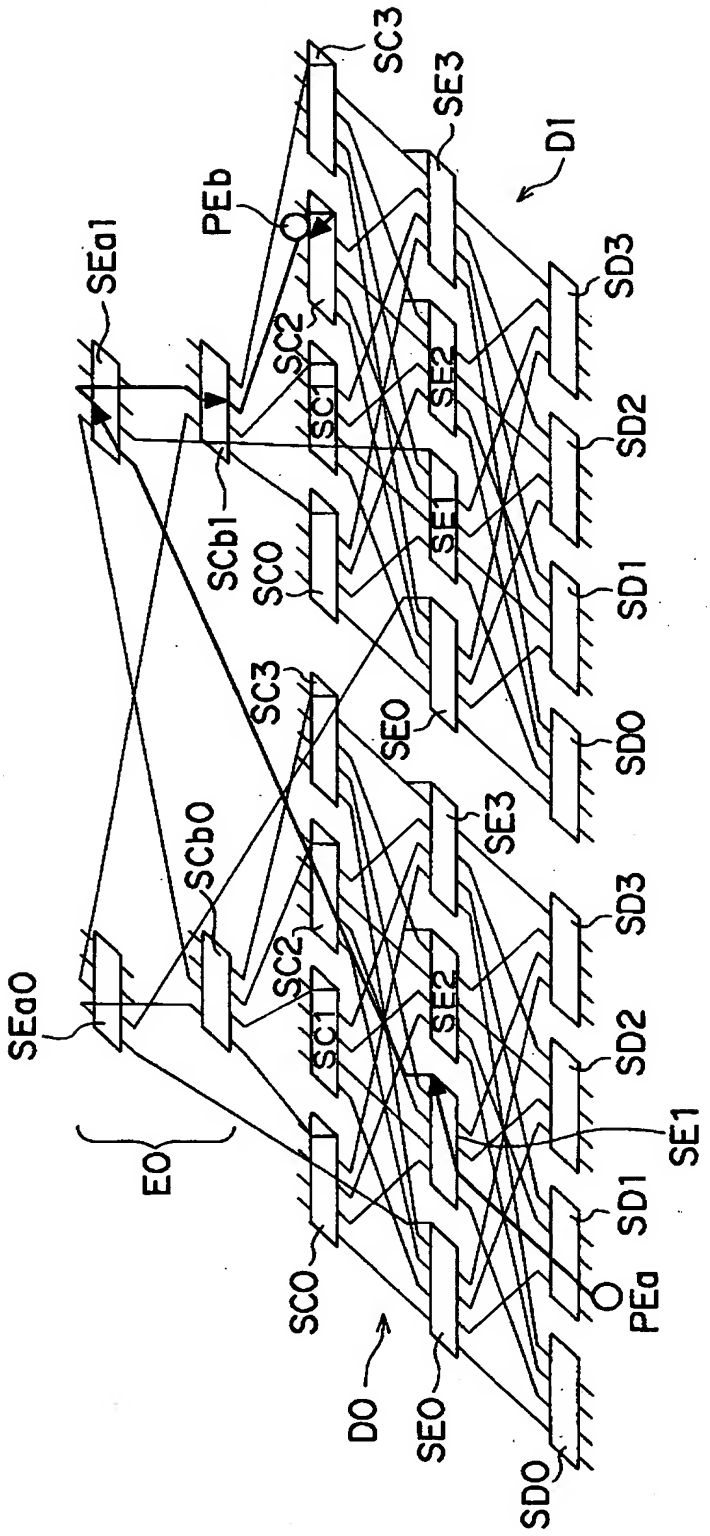
【図２１】


1a
EO
(ダミー用)
SEa3
SEa2
SEa1
SEa0
SCb3
SCb2
SCb1
SC3
SCbO
SC2
SC1
DO
SCO
SEO
D3
D2
D1
SE3
SD3
SE2
SD2
SE1
SD1
SDO

【図２２】



【図２３】


SC3
SE3
D1
PEb
SEa1
SD3
SC2
SC1
SE2
SD2
SCb1
SC0
SE1
SD1
SC3
SE0
SD0
SCb0
SE3
SD3
SEa0
SC2
SC1
SE2
SD2
SE1
E0
SD1
SC0
PEa
D0
SE0
SD0

【書類名】　要約書

【要約】

【課題】　コンパイラが容易に静的スケジューリングを行うことができ、一般的な同時アクセスパターンに対して無衝突なパケット転送を実現することができるマルチプロセッサシステム装置を得る。

【解決手段】　各プロセッサエレメント間を、階層構造の多段結合網で接続し、該多段結合網を構成する各スイッチエレメントに対して、あらかじめコンパイラによって静的にスケジューリングを行い、階層構造の多段結合網を無衝突でエミュレーションするようにした。更に、階層構造の多段結合網の基本網にクロス網を使用して１つのクロス網内でパケット転送を行う場合、レベル１のエクスチェンジャのスイッチエレメントＳＥ０～ＳＥ３に対するスケジューリングを行った際、調停に負けたパケットをスイッチエレメントＳＥ０～ＳＥ３の他のスイッチエレメントにおける空きスイッチを使用して転送するようにようにした。

【選択図】　図６

出　願　人　履　歴　情　報

識別番号　　〔３９６０２３９９３〕

１．変更年月日　　１９９６年１０月２８日
［変更理由］　新規登録
住　所　東京都港区新橋６丁目１６番１０号
氏　名　株式会社半導体理工学研究センター

２．変更年月日　　２００１年　３月２３日
［変更理由］　住所変更
住　所　神奈川県横浜市港北区新横浜３丁目１７番地２　友泉新横浜ビル６階
氏　名　株式会社半導体理工学研究センター